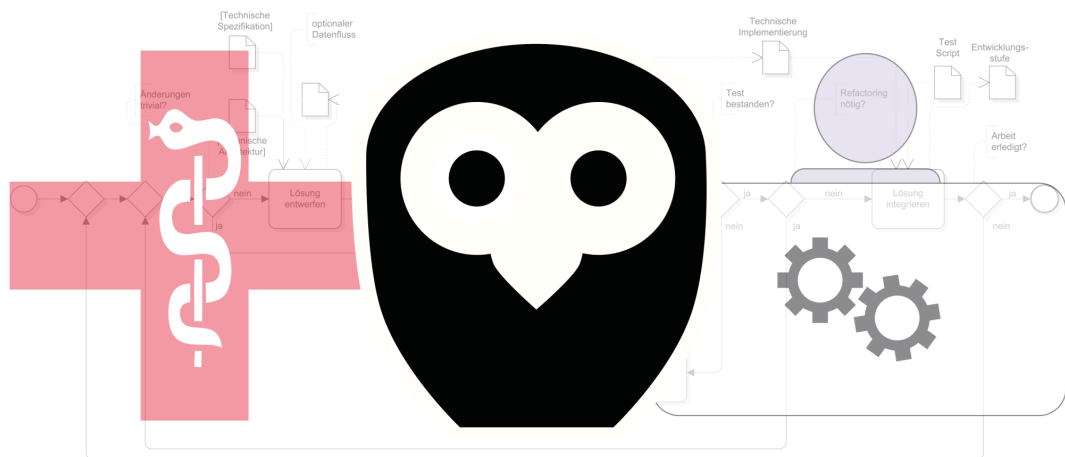




Erweiterung von BPMN um kontextuelle Einflüsse auf Prozesse abzubilden

Masterarbeit an der Universität Ulm



Vorgelegt von:

Andreas Eichwald
Andreas.Eichwald@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert
Dr. Vera Künzle

Betreuer:

Gregor Grambow

2014

Abstract

Die Welt in der wir leben unterliegt einem stetigem Wandel, welcher vor keinem Bereich halt macht. Die Notwendigkeit auf diesen Wandel adäquat und schnell zu reagieren ist nicht neu. Ebenso bekannt ist das Konzept des Prozessmanagements. Auch hier ist die Notwendigkeit auf Veränderungen zu reagieren gegeben. Sie wird durch Schlagworte wie *Flexibilität* und *Exception-Handling* gut beschrieben und auch abgedeckt. Ein Einfluss welcher sich stets ändern kann ist der Kontext. Gemeint ist hier nicht zwangsläufig die Umgebung des Prozesses sondern weitaus mehr. Die Nutzung dieses Einflusses zur Verfeinerung oder automatisierten Generierung von Prozessen ist ein Thema mit hoher Aktualität.

Ebenso aktuell ist der Siegeszug der BPMN als Prozessmodellierungs- und auch Prozessausführungssprache. Wie stellt man jedoch Kontext, kontextuelle Einflüsse und die Auswirkungen dieser dar?

In der vorliegenden Arbeit wurden Prozesse aus den Anwendungsdomänen Medizin und Software Engineering betrachtet und analysiert. Aus den Schlüssen die daraus gezogen werden konnten, wurde ein Konzept entwickelt, welche die bereits erwähnte Darstellung mittels BPMN ermöglicht.

Danksagung

Prof. Dr. Manfred Reichert für seine hilfreiche und zielführende Kritik. Des weiteren auch dafür, dass Prof. Reichert mich zum Fachbereich Medizin gebracht hat und mir somit eine unglaublich vielfältige und interessante Welt eröffnet hat.

Prof. Dr. Peter Dadam dessen Vorlesung meine Begeisterung für das Thema Prozess-management geweckt hat und somit mein Studium auf ein Ziel ausgerichtet hat.

M. Sc. Gregor Grambow für seine Korrekturen, seine Tipps und auch dafür, dass ein Termin auch sehr kurzfristig vereinbart oder abgesagt werden konnte.

B. Sc. Sabine Wieluch für die grafische Umsetzung der Idee für das Symbollogo.

B. Sc. Jonathan Sondershaus für seine Mithilfe bei der Erstellung des Titelbildes.

Melanie Herz für die Korrektur der teils äußerst diffusen Sätze und der kreativen Kommasetzung. Und auch dafür, dass sie stets ein offenes Ohr hatte, wenn ich meine Gedanken mit jemanden teilen musste, um sie zu sortieren.

Meiner Familie für den Rückhalt und Beistand während meines Studiums.

Den Fachanwendern aus den Betrachteten Domänen dafür, dass sie eine Fülle von Fragen beantwortet haben und Inspirationen für kontextuelle Einflüsse geliefert haben. Außerdem dafür, dass ich sie bei ihrer täglichen Arbeit begleiten durfte. Ohne ihre Mithilfe wäre diese Arbeit weit weniger fundiert ausgefallen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufgabenstellung	2
1.3	Aufbau der Arbeit	3
2	Grundlagen	5
2.1	Was ist BPMN	5
2.1.1	Entwicklung	5
2.1.2	Elemente der BPMN	6
2.1.3	Gründe für die Verwendung von BPMN	10
2.2	Kontext	12
3	Anforderung an die Modellierung	15
3.1	Allgemeine Anforderungen	15
3.2	Anforderungen an die Erweiterung	16
4	Methodik und Vorgehensweise	19
5	Erweiterung der BPMN	21
5.1	Prozesse in der Medizin	21
5.1.1	Modellierung ohne Kontextabhängigkeit	22
	Narkoseeinleitung (Prä-operative Phase)	22
	Intraoperative Phase und postoperative Phase	33
	Analyse	34
5.2	Prozesse im Software Engineering	36
5.2.1	Modellierung ohne Kontextabhängigkeit	37
	Der Open Unified Process	37
	Der Scrum-Prozess	39
5.3	Diskussion möglicher Alternativen	43

5.3.1	Datenobjekte und Datenfluss	44
5.3.2	Exklusive Gateways	45
5.3.3	Inklusive Gateways	45
5.3.4	Komplexe Gateways	46
5.3.5	Ereignisbasierte Gateways	47
5.3.6	angeheftete Ereignisse	52
5.3.7	benutzerdefinierte Artefakte	55
5.3.8	Kombinationen aus verschiedenen Alternativen	55
5.4	Vorschlag für neue Symbole	56
5.5	Remodellierung mit neuen Symbolen	62
5.5.1	Medizin	62
	Narkoseeinleitung (Präoperative Phase)	62
	Intra- und Postoperative Phase	67
5.5.2	Software Engineering	70
	Open Unified Process	70
	Scrum	72
6	Validierung	75
7	Verwandte Arbeiten	77
8	Zusammenfassung, Fazit und Ausblick	79
8.1	Zusammenfassung und Fazit	79
8.2	Ausblick	80
	Literaturverzeichnis	83

Seit die BPMN 2013 zum internationalen Standard erhoben wurde befindet sie sich in allen Fachbereichen auf dem Siegeszug. Die Möglichkeit Prozesse für alle Fachbereiche eines Unternehmens anwendergerecht darzustellen und diese auch automatisiert ausführen zu können hebt die BPMN ab von reinen Modellierungssprachen. Daher ist es nicht verwunderlich, dass die BPMN für beinahe jede Facette eines Prozesses ein Symbol oder eine Kombination dieser besitzt. Antrieb für die Entwicklung der Sprache ist hier, wie auch bei anderen, die Anforderung durch die Anwender.

Eine Facette von Prozessen ist der Kontext und sein Einfluss auf den Prozessverlauf. Seine Bedeutung ist sowohl Fachanwendern als auch Forschern nicht neu. Gerade in der Informatik sind Anwendungen mit Kontextbezug zu finden. Auch gibt es bereits Verknüpfungen von Kontext und Prozessmanagement [GOR10]. Die Kombination aus BPMN und Kontext ist jedoch noch offen.

1.1 Motivation

Die BPMN ist eine Prozessmodellierungssprache mit einer Vielzahl an Möglichkeiten und einer umfangreichen Symbolpalette. Auch existiert eine hohe Dichte an unterschiedlichen Designpattern für die meisten Aspekte eines Geschäftsprozesses. Ein Aspekt, dessen Modellierung bisher nicht ohne weiteres möglich ist, ist der Kontext. Das dies allerdings von Interesse ist, lässt sich gut mittels eines Beispiels herleiten. Gut geeignet hierfür ist etwas bekanntes, womit jeder bereits zu tun hatte. Die Diagnoseerstellung durch einen Arzt.

Um eine gute und genaue Diagnose erstellen zu können, muss der Arzt alle Faktoren berücksichtigen, welche Einfluss darauf haben. Eine Auswahl ist in Abbildung 1.1 zu finden. Für eine bessere Übersichtlichkeit der Darstellung wurden die Einflussfaktoren in drei Kategorien gegliedert, wobei jede Kategorie eine *Quelle* der Einflüsse darstellt.

Eine Modellierung dieses Beispiels in BPMN würde zu einem unübersichtlichen Prozess führen, da neben den aufgezeigten Einflüssen weitere Faktoren eine Rolle spielen könnten. Würde man alle modellieren wollen, wäre der resultierende Prozess für einen Menschen nicht ohne lange

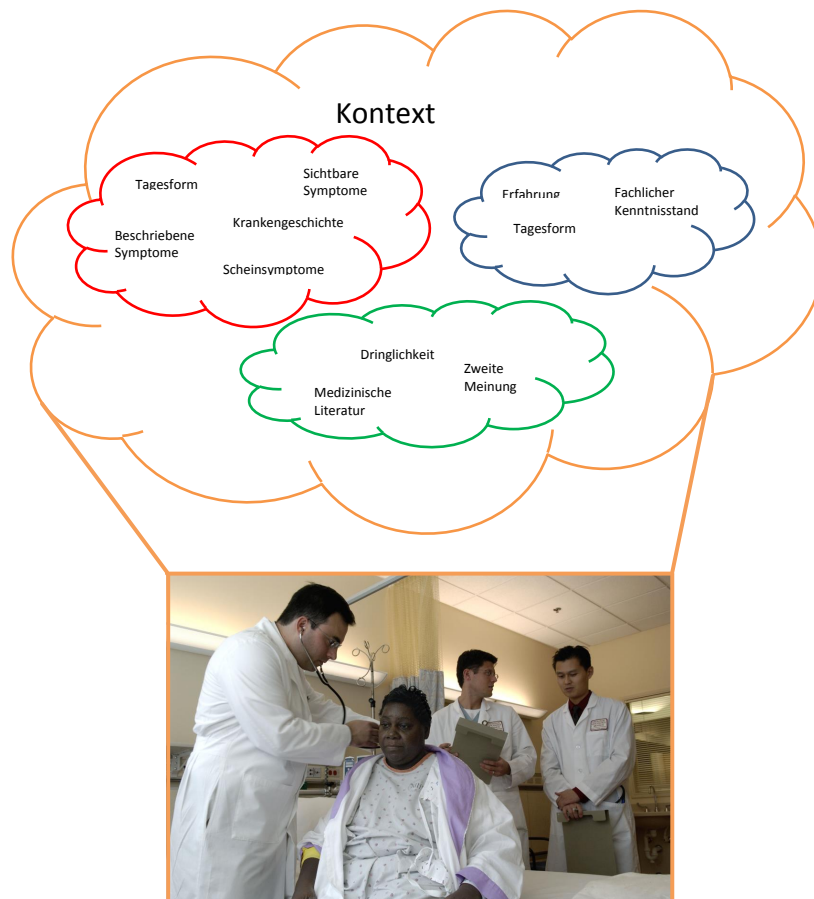


Abbildung 1.1: Kontexteinfluss bei Diagnoseerstellung, Quelle: National Cancer Institute und eigene Bearbeitung

Einarbeitungszeit oder hohes Fachwissen verständlich. Solche Prozessmodelle sollen jedoch häufig auch für Fachfremde zugänglich und verständlich sein, um diesem damit komplexe Sachverhalte und Abläufe näher zu bringen und zu erläutern. Gerade dies ist eine Stärke der BPMN, welche auch für die Modellierung von Kontext eine solche bleiben soll.

1.2 Aufgabenstellung

Ziel der vorliegenden Arbeit ist es zu untersuchen wie Kontext in Prozessen mithilfe der BPMN modelliert werden kann. Hierfür werden Prozesse auf zwei unterschiedlichen Fachbereichen aufgenommen, modelliert und anschließend auf ihre kontextuellen Einflüsse analysiert. Des weiteren muss untersucht werden ob es Möglich ist Kontext mit den Symbolen der BPMN darzustellen ohne die Modelle stark zu verkomplizieren. Sollte dies nicht möglich sein, müssen die Gründe hierfür beschrieben und erläutert werden. Zudem soll ein Vorschlag für die Erweiterung der BPMN gemacht werden, welcher die Modellierung von Kontext möglich macht. Sofern ein solcher Vorschlag gemacht wird, muss dieser beschrieben und die Verwendung anhand von Bei-

spielen erklärt werden. Des weiteren müssen die Prozesse, welche zuvor aufgenommen wurden unter Verwendung dieses Vorschlags neu modelliert werden.

Abschließend muss der Vorschlag, sowohl gegen die Aufgabenstellung als auch gegen zuvor festgelegte Anforderungen validiert werden. Sollte eine Anforderung oder ein Punkt der Aufgabenstellung nicht erfüllbar sein, so ist dies zu begründen und zu beschreiben.

1.3 Aufbau der Arbeit

Die Arbeit gliedert sich in fünf Kapitel. Den Anfang macht das Kapitel zwei, wo die für die Arbeit benötigten Grundlagen erläutert werden. Hierbei wird zuerst auf das Thema BPMN und ausgewählte Elemente der Sprache eingegangen und im Anschluss der Kontextbegriff erläutert. Mittels dieser Erläuterungen wird eine mögliche Definition für Kontext im Prozessmanagement aufgebaut. Das darauf folgende Kapitel 3 befasst sich mit den Anforderungen an die in der Arbeit verwendete Modellierung. Diese werden in zwei Teile unterteilt, wobei der erste Teil die allgemeinen Anforderungen an die Modellierung beschreibt, während der Zweite die Anforderungen an eine mögliche Erweiterung zusammenfasst. Als letztes konzeptionelles Kapitel (Kapitel 4) vor dem Kernteil der Arbeit wird die Methodik und Vorgehensweise der Arbeit umrissen.

Im Kernteil, zu finden in Kapitel 5, sind Beschreibungen zu den betrachteten Anwendungsdomänen und die dazugehörenden Prozesse samt Analysen zu finden. Auf diese folgt die Diskussion möglicher alternativer Modellierungen unter Verwendung BPMN eigener Mittel. Nach Abschluss dieser erfolgt die Beschreibung des Vorschlags für neue Symbole mitsamt zugehöriger Beispiele.

Die zuvor genannten Schritte waren die Hinführung und Beschreibung eines Vorschlags zur Erweiterung der BPMN. Dieser Vorschlag wird in dem darauf folgenden Bereich auf seine Praxis-tauglichkeit hin geprüft, indem die zuvor beschriebenen Prozesse neu modelliert werden. Diese Modellierungen werden darauffolgend ebenfalls analysiert, was zudem den Schluss des Kern-teils bildet.

Den Abschluss der vorliegenden Arbeit, zu finden in Kapitel 6, bilden eine kurze Zusammenfassung sowie ein Ausblick und das Fazit.

Im Folgenden Kapitel wird auf die für die vorliegende Arbeit notwendigen Grundlagen eingegangen. Begonnen wird mit einer allgemeinen Einführung in das Konzept d sogenannten BPMN und der Beschreibung der im Folgenden verwendeten Symbole. Im weiteren Verlauf wird der Kontextbegriff eingeführt und erläutert. Mittels dieser Erläuterung und Beispielen für die Verwendung von Kontext in der Informatik werden angepasste Definitionen abgeleitet. Diese bilden den Grundstein für die Analysen in den darauf folgenden Kapiteln.

2.1 Was ist BPMN

BPMN wurde 2004, maßgeblich von IBM, entwickelt. Die Abkürzung bedeutete in der damaligen Revision *Business Process Modeling Notation* und stand stellvertretend für eine Menge von Symbole und deren Verknüpfungsregeln.

2.1.1 Entwicklung

Die Urform von BPMN eignete sich lediglich zur visuellen Darstellung von Prozessmodellen. Diese konnten anschließend, zum Beispiel in einer Präsentation, verwendet werden oder zu Dokumentationszwecken von Prozessen genutzt werden. Hier liegt auch einer der entscheidenden Vorteile von BPMN: die damit erstellten Prozessmodelle können von jedem intuitiv gelesen und verstanden werden.

Im Jahr 2005 wurde die BPMN von der *Object Management Group* (OMG), siehe hierzu [OMG], übernommen und weiterentwickelt. Eine der wichtigsten Neuerungen war die automatisierte Prozessausführung. Diese Möglichkeit existiert erst seit der 2011 verabschiedeten Version 2.0. Im Zuge dieser neuen Revision wurde auch der Name, welcher sich hinter dem Kürzel verbirgt, in *Business Process Model and Notation* geändert. Die neue Revision erhöhte die Mächtigkeit von BPMN durch Einführung neuer Konzepte wie die Ausführungssemantik, die Beschreibung von Prozessanwendern und der Möglichkeit die BPMN zu erweitern.

2.1.2 Elemente der BPMN

Der Sprachumfang von BPMN lässt sich prinzipiell in fünf Gruppen einteilen, diese werden Basiselemente genannt. Jede von ihnen beschreibt eine Menge von Elementartypen. Im Folgenden werden diese Basiselemente benannt und kurz beschrieben. Auf Elemente, welche im Verlauf der Arbeit Verwendung finden, wird genauer eingegangen. Eine detaillierte Beschreibung aller BPMN Elemente ist bei [OMG11] zu finden.

Flussobjekte

In diesem Basiselement sind die Aktivitäten, die Gateways und die Ereignisse beschrieben. Wobei Aktivitäten die auszuführenden Prozessschritte definieren, welche durch die Gateways verknüpft sind und zu Ereignissen führen. Abbildung 2.1 zeigt ein einfaches Beispiel für die Verwendung von Flussobjekten.

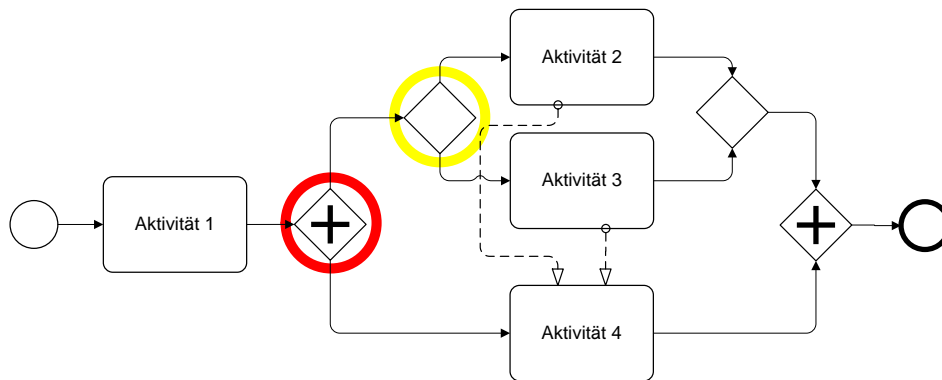


Abbildung 2.1: Beispiel für Flussobjekte und Verbindende Objekte

Die erste Untergruppe, welche hier beschrieben wird, sind die Gateways. In der vorliegenden Arbeit werden mehrere Gateway-Arten verwendet, diese sind:

- datenbasierte Gateways (exklusiv und parallel)
- Ereignis-basierte Gateways (exklusiv, parallel, „normal“)
- inklusive Gateways
- komplexe Gateways

Eine anschauliche Möglichkeit die Funktion eines bestimmten Gateway zu erklären ist das Token Prinzip, welches hier exemplarisch anhand von Abbildung 2.1 erläutert wird.

Das Token wird am *Startereignis* erschaffen, dies ist das am weitesten links stehende Flussobjekt. Es wird automatisch an Aktivität 1 weitergereicht und löst die Ausführung dieser aus. Nach Abschluss der Aktivität wandert das Token zu dem rot markierten Gateway. Dieses ist ein paralleles Gateway, was bedeutet, dass das Token geklont wird und beide Pfade gleichzeitig abläuft. Die Pfade eines parallelen Gateways müssen nicht beschriftet werden.

Auf dem oberen Pfad trifft das Token auf ein exklusives Gateway (gelbe Markierung). Im Gegensatz zu dem parallelen Gateway kann hier nur eine der beiden Aktivitäten (2 oder 3) ausgeführt werden, da nur einer der möglichen Pfade durchlaufen werden kann. Dies ist auch der wichtigste Unterschied zwischen diesen beiden Gateway-Typen.

Die zwei zuvor beschriebenen Elemente gehören zu den datenbasierten Gateways. Sie werden durch Daten des Prozesses gesteuert. Diese Daten werden vom Gateway ausgewertet. Abhängig von Typ des Gateways werden eine, bei einem exklusiven Gateway, oder mehrere, bei parallelen oder inklusiven Gateways, Aktivitäten ausgeführt.

Eine etwas andere Ausführungssemantik haben die Ereignis-basierten Gateways. Sie werten Ereignisse aus, welche dem Gateway nachgeschaltet sind. Sobald eines der Ereignisse eingetreten ist, werden alle anderen Prozesspfade gesperrt.

Einschub: Gateway-Bezeichnungen

Gateways werden im meist als Paar verwendet, diese werden als *Split* und *Join* bezeichnet. Als Split bezeichnet man ein Gateway, an welchem der Prozessfluss in mehrere Pfade aufgeteilt wird. Auf ein Split-Gateway folgt in den meisten Fällen das korrespondierende Join-Gateway. An diesem werden die zuvor aufgeteilten Prozesspfade wieder zusammengeführt. Je nach Modellierungssprache ist ein solcher Join verpflichtend oder optional. In der BPMN ist ein Join-Gateway optional.

Das letzte Gateway, welches hier beschrieben wird, ist das sogenannte *komplexe* Gateway. Es steht für die Vereinigungs- (Join) und Verzweigungsverhalten (Split), welche nicht von den anderen Gateways ausgeführt werden können. In der vorliegenden Arbeit wird es lediglich als Join-Gateway verwendet. Es ermöglicht unter anderem die Auswahl von N Prozesspfaden aus M möglichen.

Ein weiteres Flussobjekt, das in der vorliegenden Arbeit Verwendung findet, ist das Zwischenereignis. Dieses kommt sowohl angeheftet an eine Aktivität als auch als alleinstehendes Flussobjekt vor. Bei den angehefteten Zwischenereignissen unterscheidet man zwischen den *unterbrechenden* und den *nicht unterbrechenden* Ereignissen, ein generisches Beispiel ist in Abbildung 2.2 dargestellt. Der Unterschied liegt, wie der Name bereits vermuten lässt, in der Ausführung der Aktivität, an welche das Ereignis angeheftet wurde. Bei einem unterbrechenden angehefteten Ereignis wird die Ausführung der Aktivität, an welche das Ereignis angeheftet ist, unterbrochen und der durch das Ereignis eingeleitete Pfad wird ausgeführt. Bei einen nicht unterbrechenden Ereignis wird die Bearbeitung der Trägeraktivität nicht unterbrochen. Der durch das Ereignis eingeleitete Pfad wird parallel aktiviert und durchlaufen.

In der vorliegenden Arbeit werden beinahe alle in der BPMN definierten Ereignisse verwendet. Im einzelnen währen dies:

- Nachricht
- Zeit

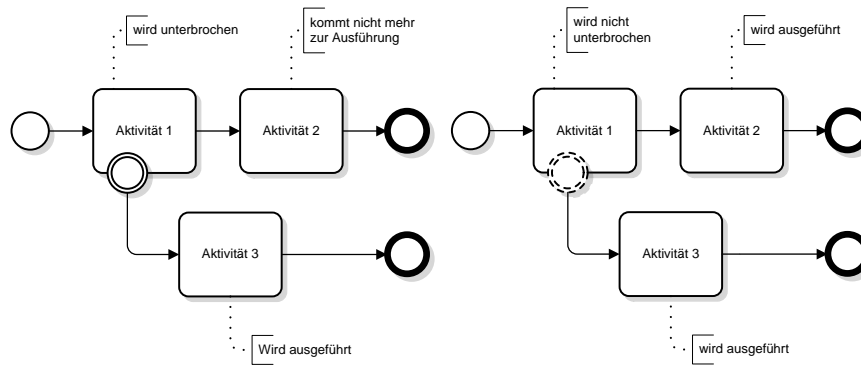


Abbildung 2.2: Beispiel für unterbrechende (links) und nicht unterbrechende Ereignisse (rechts)

- Bedingung
- Signal
- Eskalation
- Mehrfach

Den Anfang macht das Nachrichteneignis. Eine Nachricht in BPMN ist immer an einen bestimmten Adressaten gerichtet und muss nicht zwangsläufig eine Mail, ein Anruf oder ein Brief. Die Aufteilung in unterbrechend und nicht unterbrechend ist identisch zu der generischen Beschreibung.

Das Zeit Ereignis existiert sowohl als unterbrechendes als auch als nicht unterbrechendes angehängtes Ereignis. Im unterbrechenden Fall wird die Aktivität, an welche das Ereignis angehängt ist, nach Ablauf einer bestimmten Zeitspanne unterbrochen und der auf das Ereignis folgende Prozesspfad wird weiter verfolgt. Beim nicht unterbrechenden Fall kann ein Zeitpunkt vor Abschluss der Aktivität angegeben werden. Auf diese Art können, zum Beispiel vorbereitend Arbeiten für die Nachfolgenden Aktivitäten beschrieben werden.

Die wichtigste Eigenschaft des Bedingungsereignisses ist, dass die Bedingung außerhalb des Prozesses erfüllt werden muss. Auch dieses Ereignis existiert als unterbrechendes und als nicht unterbrechendes Ereignis. Wie auch beim Nachrichteneignis ist die Aufteilung identisch zur generischen Beschreibung.

Das Signal, in der BPMN, ist bis auf einen Unterschied identisch zur Nachricht, dieser ist der Adressat. Während die Nachricht immer zwingend einen Adressaten braucht, geht das Signal prinzipiell an alle. Als Adressat kann die Person gesehen werden, welche als erste auf das Signal reagiert.

Das nächste Ereignis auf der Liste ist die Eskalation. Dieses dient der Kommunikation zwischen Oberprozess und Subprozess.

Das letzte angehängte Ereignis, das in der Arbeit Verwendung findet, ist das Mehrfacher Ereignis. Es kann genutzt werden um mehrere mögliche Ereignisse in einem Symbol zusammenzufassen.

Zusätzlich zu den bereits erwähnten Ereignissen wurde für die Modellierung der vorliegen-

den Prozesse das *Link-Ereignis* genutzt. Dieses existiert ausschließlich als ausgelöstes und als eingetretenes Ereignis. Es dient rein der Strukturierung großer Prozesse und zur Aufteilung dieser in mehrere Teile. Ein Beispiel für die Verwendung dieses Ereignisses ist in Abbildung 2.3 zu sehen.



Abbildung 2.3: Beispiel für Link Ereignisse

Man kann sich das Beispiel aus der Abbildung als Ausschnitt aus einem größeren Gesamtprozess vorstellen. In diesem Fall würden die ersten n Aktivitäten auf einer Seite und die Aktivitäten ab Nummer $n+1$ auf der nächsten Seite stehen. Die Link-Ereignisse verbinden den Prozess logisch, so dass man ihn auf mehrere Teile aufsplitten kann.

Verbindende Objekte

Diese Objekte definieren den „Prozessfluss“. Sie bestimmen die Reihenfolge der Aktivitäten oder Gateways. Ebenso zeigen Verbindende Objekte den Nachrichtenfluss zwischen den Prozessinstanzen. Flussobjekte werden mit einem durchgezogenen Pfeil verbunden, wobei die Pfeilrichtung die Richtung des Prozessflusses definiert. Beim Prozessfluss gibt es eine Besonderheit, den Standardpfad. Dieser wird durch einen kleinen Querstrich am Anfang des Pfeils gekennzeichnet. Der Standardfluss wird durchlaufen, wenn alle anderen Bedingungen nicht zutreffen. Ebenso existieren Verbindende Objekte für den Nachrichtenfluss, diese werden durch einen gestrichelten Pfeil dargestellt. Wobei am Ausgang des Pfeils, welcher beim Absender angehängt ist, ein leerer kleiner Kreis befindet. Wie auch beim Prozessfluss definiert die Richtung des Pfeils die der Kommunikation. Die letzte Art der Verbindenden Objekte sind die Assoziationen, dargestellt als gepunktete Linie. Diese gibt es als ungerichtete und als gerichtete Variante. Die Erste wird verwendet um andere Objekte mit einem Kommentar zu versehen (siehe Abbildung 2.4 für ein Beispiel). Die Zweite verbindet Aktivitäten oder Gateways mit Datenobjekten. Gerichtete Assoziationen werden zusätzlich mit einer Pfeilspitze versehen, bei welcher die Linien nach innen abgerundet sind. Wie auch bei den zuvor beschriebenen Objekten, definiert die Pfeilrichtung die Flussrichtung.

Artefakte

Diese Basiselemente dienen der Strukturierung, mittels Gruppen, und Erläuterung innerhalb des Modells durch Anmerkungen. Des Weiteren erlaubt der Standard hier das Hinzufügen eigener Elemente. Ein Beispiel für die Artefakte Gruppe und Anmerkung ist in Abbildung 2.4 zu finden.

Teilnehmer

Die Teilnehmer eines Prozesses, sprich diejenigen, welche Prozessschritte ausführen, werden in der BPMN-Standard als Swimlanes dargestellt. Diese Lanes werden in einem

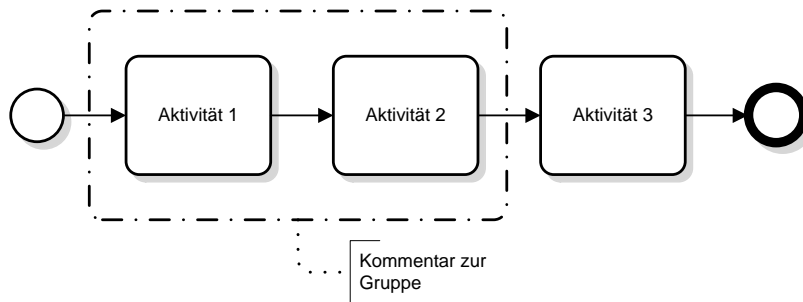


Abbildung 2.4: Beispiel für Artefakte

Pool zusammengefasst, welcher das Bezugssystem darstellt, in welchem die Teilnehmer agieren. Ein Beispiel hierfür ist in Abbildung 2.5 zu sehen.

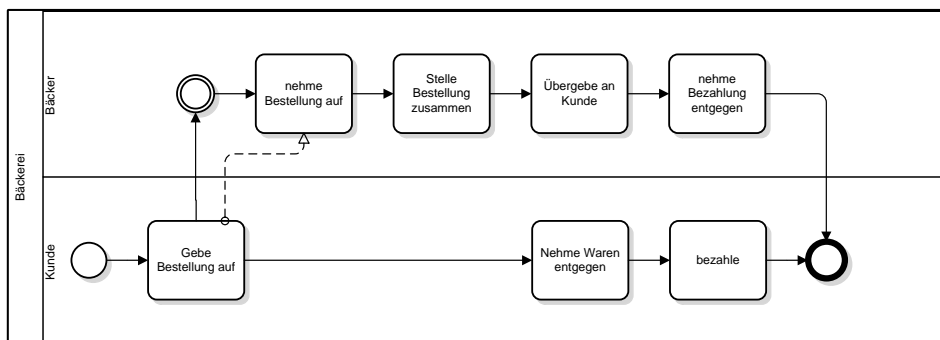


Abbildung 2.5: Beispiel für Teilnehmer

Daten

Ein integraler Bestandteil eines Prozesses ist der Datenfluss. Der BPMN Standard definiert mehrere unterschiedliche Datenelemente, von denen jedoch nur eines in der Arbeit Verwendung findet. Ein Beispiel für die Nutzung dieser Elemente wird in Abbildung 2.6 gegeben.

Dieser Abschnitt lieferte eine kleine Übersicht über die vom Standard definierten Symbole, weitere Informationen über diese und ihre Verwendung sind bei [FR12], [All09] und [Sil09] zu finden.

2.1.3 Gründe für die Verwendung von BPMN

Es gibt viele Gründe die für die Verwendung von BPMN zur Prozessmodellierung sprechen. Allen voran die Tatsache, dass BPMN ein anerkannter und weit verbreiteter Standard ist und somit in vielen Unternehmen eingesetzt wird. Der Symbolsatz ist intuitiv verständlich und leicht zu erlernen, beschränkt man sich auf die Basiselemente. Man kann damit bereits einfache Ge-

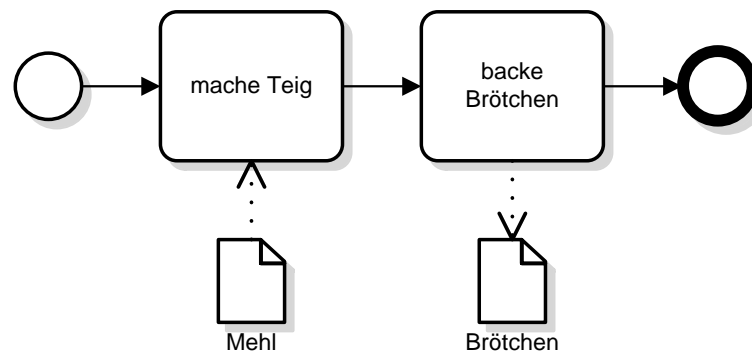


Abbildung 2.6: Beispiel für Datenobjekte

schäftsprozesse schnell modellieren, ein Beispiel hierfür ist in Abbildung 2.7 zu sehen. Dort wird ein einfaches Beispiel aus der Anwendungsdomäne Logistik aufgezeigt. Der Prozess beginnt mit dem Entgegennehmen einer Lieferung und anschließender Prüfung auf Vollständigkeit. Die Aktivität *nehme Lieferung entgegen* wird von einem Mitarbeiter des Lagers ausgeführt. Dieser benachrichtigt bei Erhalt der Lieferung einen Mitarbeiter aus der Buchhaltung, so dass dieser die Rechnung prüfen kann. Zeitgleich prüft der Lagermitarbeiter die Lieferung auf Vollständigkeit. Falls diese vollständig ist, wird wiederum die Buchhaltung benachrichtigt, damit die Rechnung beglichen werden kann. Sollte jedoch etwas fehlen wird eine Meldung an den Lieferanten gemacht. Unabhängig davon ob die Lieferung vollständig war oder nicht wird diese in das Lager einsortiert und der Prozess des Lagers endet damit. Ähnlich verhält es sich für die Buchhaltung. Nach Begleichen der Rechnung wird die ebenfalls einsortiert und der Prozess endet. Bei diesem Prozess handelt es sich um ein Minimalbeispiel, welches keinen Anspruch auf Vollständigkeit erhebt.

Der zuvor beschriebene Prozess wurde mit einem Minimum an Symbolen erstellt. Da das Beispiel sehr einfach gehalten ist war diese Minimalauswahl ausreichend. Auch komplexere und größere Prozesse lassen sich auf diese Art darstellen. Jedoch ist es gerade bei komplexeren Prozessen häufig hilfreich mehr Symbole zu verwenden, da diese oft eine kompaktere Modellierung ermöglichen. Ein weiterer Grund der für die Verwendung von BPMN spricht, ist die Vielzahl von Editoren. Es gibt sowohl kostenpflichtige als auch kostenlose Varianten, so dass jeder den für sich passenden finden kann. Eine Übersicht über verfügbare Tools ist unter [Too] zu finden. Die hier verwendeten Prozesse wurden mit Microsoft Visio 2010 erstellt.

Der letzte hier zu nennende Vorteil von BPMN ist die Möglichkeit den selben Prozess in unterschiedlicher Granularität schnell und mit dem gleichen Tool zu erstellen. Dies ist gerade in den Fällen von Bedeutung, wenn der Prozess für mehrere Personen in unterschiedlichen Positionen dargestellt werden soll. Für das Management ist das letzte Ausführungsdetail weniger von Bedeutung als zum Beispiel für die IT, welche den Prozess technisch umsetzen soll.

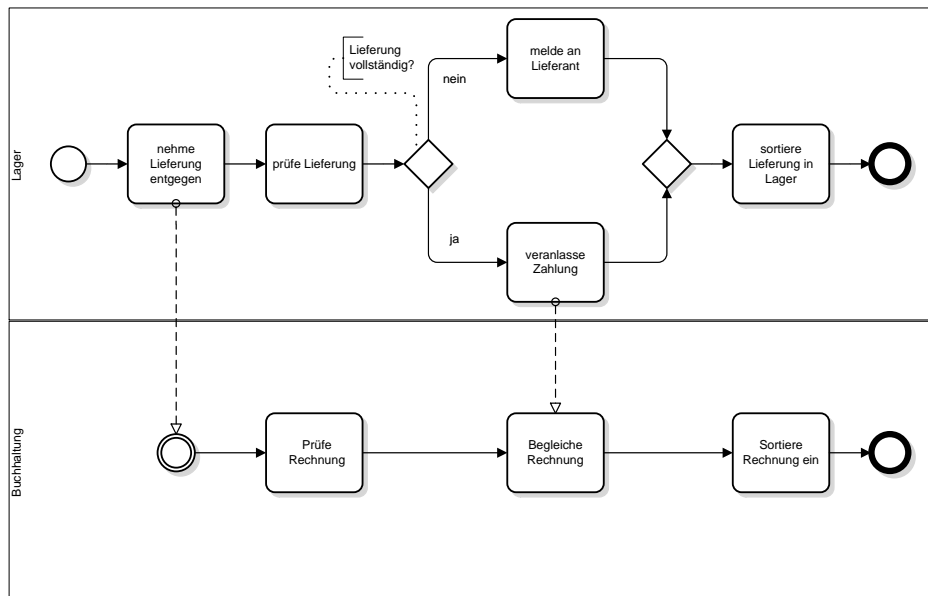


Abbildung 2.7: Einfaches Lagerbeispiel

2.2 Kontext

Um kontextuellen Einfluss auf Prozesse zu erkennen und eine adäquate Modellierung anzubieten, muss man verstehen was Kontext genau bedeutet. Der Duden definiert Kontext folgendermaßen (Definition 2.1):

Definition 2.1 Kontext

1. *umgebender Text einer sprachlichen Einheit*
2. *(relativ selbstständiges) Text- oder Redestück*
3. *inhaltlicher Gedanken-, Sinnzusammenhang, in dem eine Äußerung steht, und Sach- und Situationszusammenhang, aus dem heraus sie verstanden werden muss*
4. *Zusammenhang*

Betrachtet man diese Definition stellt man fest, dass die meisten Aussagen sich auf die gesprochene oder geschriebene Sprache beziehen. Einzig der letzte Punkt eröffnet Interpretationsspielraum. Sieht man den Kontext als Zusammenhang zwischen möglichen bzw. erfolgten Aktionen und dem Bezugsrahmen, in welchem diese erfolgen, ist es möglich eine Brücke zu informatiklastigen Themen zu schlagen und den Begriff Kontext sinnvoll zu verwenden.

In der Informatik ist Kontext vor allem im Ubiquitous Computing unter dem Schlagwort *Context Aware Computing* zu finden (siehe hierfür [ADB⁺99]). Dies bedeutet das Anpassen der konkreten Nutzung einer allgemeinen Anwendung an die jeweilige lokale Situation. Hierfür wird eine

Anwendung um den Kontextbezug erweitert. Die Erweiterung umfasst hierbei zwei Aufgabenstellungen:

1. Das System erhält die Fähigkeit Aspekte seiner Umwelt zu erfassen.
2. Die Fähigkeit auf diese zu reagieren.

Ein Beispiel für Context Aware Computing ist die Nutzung von Ortsinformationen zur Darstellung passender Informationen. Es ist jedoch genauso möglich den gesundheitlichen Zustand des Nutzers als Kontext zu nutzen, um beispielsweise automatisch den Arzt zu verständigen, wenn der Puls des Nutzers unter einen vorher definierten Grenzwert fällt.

Wendet man diese Erkenntnisse auf das Prozessmanagement an sieht man schnell, dass hier die selbe Erweiterung der Modellierung nötig ist wie beim Context Aware Computing. Fasst man das bisher Genannte zusammen und kombiniert es mit 2.1 lässt sich daraus eine Definition des Begriffs Kontext im Rahmen des Prozessmanagement formulieren. Eine solche finden sie in Definition 2.2.

Definition 2.2 *Kontext im Prozessmanagement*

ist die Gesamtheit der für den Prozess relevanten Umgebungsvariablen und -ereignisse, deren Änderung bzw. Eintreten oder Ausbleiben den Prozessablauf beeinflussen/ verändern können.

Für eine detailliertere Modellierung von kontextuellen Einflüssen ist es hilfreich zwei zusätzliche Begriffe einzuführen. Diese erlauben eine feingranulare Abstimmung von Modell und Realität. Dies wäre zum einen der Begriff kontextsensitiv und zum anderen kontextabhängig. Um eine formale Definition zu finden, bietet der Duden eine solide Basis. Dieser definiert den Begriff wie folgt:

Definition 2.3 *kontextsensitiv nach Duden*

(in Bezug auf das Vorkommen eine sprachlichen Einheit) vom Kontext abhängig, auf einen Kontext beschränkt

Diese rein linguistische Definition lässt sich allerdings nicht ohne weiteres auf Prozessmanagement anwenden. Zudem geht aus der Definition hervor, dass die Begriffe *kontextsensitiv* und *kontextabhängig* Synonyme sind. Das selbe Ergebnis liefert eine Kontrollabfrage bei *openThesaurus.org*. Aus diesen Gründen muss auch hier, wie bereits zuvor beim Kontext allgemein, eine angepasste Definition erstellt werden.

Definition 2.4 *kontextsensitiv (in Bezug auf Prozesse):*

Eine Prozessaktivität ist kontextsensitiv, wenn diese auf sich verändernden Kontext reagiert. Hierfür muss die Aktivität nicht abgebrochen werden.

Wie man an Definition 2.4 erkennen kann, baut diese auf der des Dudens (Definition 2.3) auf. Ebenso verhält es sich mit der nachfolgenden Definition für Kontextabhängigkeit, allerdings spielt bei dieser auch die Beschreibung des Context Aware Computing eine wichtige Rolle.

Definition 2.5 *kontextabhängig (in Bezug auf Prozesse):*

Eine Prozessaktivität ist kontextabhängig, wenn die Aktivität durch sich ändernden Kontext gestartet, gestoppt oder in ihrer Ausführung verändert wird.

Im Folgenden werden diese Definitionen genutzt, um eine Erweiterung der BPMN zur Modellierung der Kontextabhängigkeit zu erarbeiten. Auf den folgenden Seiten werden die Anforderungen an solch eine Modellierung diskutiert und die Vorgehensweise bei der Erarbeitung beschrieben.

Anforderung an die Modellierung

Die Anforderung an die Modellierung wird in zwei Bereiche unterteilt. Der Erste umfasst die allgemeinen Anforderungen an die Modellierung der untersuchten Prozesse und der Beispiele, während der Zweite sich den Anforderungen an die für den Kontexteinfluss verwendeten Symbolen und den daraus resultierenden Modellen ergibt und als Erweiterung des Ersten gesehen werden kann.

3.1 Allgemeine Anforderungen

Eine der wichtigsten Anforderungen ist die Vollständigkeit. Bei der Modellierung ist zu beachten, dass die Prozessmodelle ausreichend Informationen beinhalten um den dem Modell zugrundeliegenden Vorgang vollständig zu dokumentieren. Demnach muss zunächst geklärt werden, ob jeder notwendige Aspekt mittels des Symbolsatzes der BPMN modelliert werden kann. Zur Modellierung können alle Elemente der BPMN verwendet werden, mit Ausnahme der eigenen Artefakte. Grund hierfür ist, dass diese nicht in allen BPMN Modellierungswerkzeugen verwendet werden können.

Ein ebenso wichtiges Kriterium für die Modellierung ist die Korrektheit. Hierbei muss allerdings zwischen syntaktischer und semantischer Korrektheit unterschieden werden. Die syntaktische Korrektheit fordert, dass die Prozessmodelle nicht gegen die Regeln der BPMN 2.0 Syntax verstoßen. Diese sind unter [OMG11] nachzulesen.

Die semantische Korrektheit hingegen fordert, dass die Modelle die Realität korrekt widerspiegeln und in sich geschlossen sind. Dies ist notwendig um Widersprüche oder Unklarheiten in den Modellen zu vermeiden.

Des weiteren besteht die Anforderung an die Verständlichkeit der Prozessmodelle. Die Prozesse sind stets so zu modellieren, dass diese von einem menschlichen Benutzer gelesen und verstanden werden können. Hierfür muss der Detailgrad des Modells korrekt gewählt werden. Das Modell muss ausreichend detailliert sein um den Prozess zu verstehen, jedoch nicht so detailliert um diesen zu überladen und damit die Lesbarkeit und Verständlichkeit zu reduzieren.

Der Prozess soll so gestaltet sein, dass sowohl ein Fachanwender als auch jemand der sich nicht im Fachbereich auskennt, diesen verstehen kann. Die Fachanwender sollen den, dem Modell zugrunde liegenden, Prozess identifizieren und validieren können. Anwender, welche sich nicht im Fachbereich auskennen, sollen ohne lange Einarbeitungszeit den Prozess verstehen und zusammenfassen können.

Die zuvor genannten Anforderungen basieren auf den Grundsätzen ordnungsgemäßer Modellierung nach [Rei13]. Diese sind:

- Vergleichbarkeit
- Systematischer Aufbau
- Richtigkeit (Syntax und Semantik)
- Relevanz
- Wirtschaftlichkeit
- Klarheit

3.2 Anforderungen an die Erweiterung

Wie zu Beginn erwähnt, handelt es sich bei diesen Anforderungen um eine Erweiterung der allgemeinen Anforderungen. Diese sind notwendig um eine sinnvolle und nutzbare Erweiterung der BPMN erstellen zu können.

Die Erweiterung der BPMN muss syntaktisch widerspruchsfrei sein. Dies soll heißen, dass sie nicht der Anforderung A2 (siehe Tabelle 3.1) widerspricht. Des weiteren muss die Erweiterung sich in die bestehende Struktur der BPMN einordnen lassen.

Ebenso muss die Erweiterung syntaktisch neuartig sein. Die veränderte Verwendung bereits bestehender Symbole gilt nicht als Erweiterung und erfüllt somit nicht die Zielsetzung der Arbeit. Eine Überschneidung in der Semantik zwischen der Erweiterung und den Bestehenden Symbolen ist erlaubt.

Zudem muss die Erweiterung die Modellierung kontextueller Einflüsse auf Prozesse darstellbar machen. Es muss sowohl die der Einfluss als auch seine Auswirkung modellierbar sein. Das Modell darf dabei nicht unnötig verkompliziert werden.

Die letzte Anforderung bezieht sich auf die Vergleichbarkeit von Modellen. Ein Prozessmodell, welches mittels der bestehenden BPMN Syntax erstellt wurde, muss semantisch mit einem Prozessmodell, welches mittels erweiterter Syntax erstellt wurde, übereinstimmen. Vorausgesetzt, dass der zugrundeliegende Prozess der gleiche ist. Abweichungen, welche sich auf Grund d Kontextbetrachtung ergeben, sind erlaubt.

Die folgende Tabelle fasst die zuvor beschriebenen Anforderungen zusammen und versieht diese mit einem Kürzel. Dieses dient der späteren Referenzierung der Anforderungen. Wobei die

Kürzel A1 bis A4 die allgemeinen Anforderungen und E1 bis E4 die Anforderung an die Erweiterung referenzieren.

Kürzel	Anforderung
A1	Vollständigkeit
A2	syntaktische Korrektheit
A3	semantische Korrektheit
A4	Verständlichkeit
E1	syntaktische Widerspruchsfreiheit
E2	syntaktische Neuartigkeit
E3	Kontextmodellierung
E4	Vergleichbarkeit

Tabelle 3.1: Anforderungen, Quelle: eigene Darstellung

Methodik und Vorgehensweise

Die Prozesse in der vorliegenden Arbeit wurden, im Rahmen von mehreren kleineren Fallstudien aufgenommen oder aus Vorgehensmodellen extrahiert. Diese beschränkten sich auf Beobachtung der Abläufe und Unterhaltungen mit Fachanwendern, um zusätzliche Details über die Prozesse zu erfahren. Für die Fallstudien in der Medizin wurden mehrere Kliniken besucht. Um möglichst viele Aspekte der Prozesse betrachten zu können, wurden hierfür mehrere Fachanwender begleitet. Die Prozesse im Software Engineering konnten nicht im Rahmen einer Fallstudie erfasst werden, da dies den Zeitrahmen dieser Arbeit gesprengt hätte. Die grafischen Repräsentationen wurden den genannten Quellen entnommen und überarbeitet. Der fachliche Hintergrund für die Prozesse stammt aus Literatur und Gesprächen mit Fachanwendern.

Ähnlich verhält es sich mit den kontextuellen Einflüssen. Die Einflüsse auf die medizinischen Prozesse konnten zum Teil während der Prozessaufnahme beobachtet werden. Weitere Einflüsse stammen von bereits erwähnten Fachanwendern. Beim Software Engineering wurden die Einflüsse zum größten Teil von Fachanwendern benannt.

Die Prozesse wurden auf ihre kontextuellen Einflüsse hin untersucht. Basis hierfür waren die in Kapitel 2 vorgestellten Definitionen. Diese Erkenntnisse flossen im Anschluss darauf in die Betrachtung der alternativen Modellierungen, zu finden im Kapitel 5.3, und in den Vorschlag für eine Erweiterung der BPMN.

Diese entstanden in mehreren Iterationen. Die ersten Fallstudien zeichneten das grobe Konzept, welches jedoch viele redundante oder unnötige Symbole enthielt. Nach weiteren zwei Iterationen war das Konzept, wie es in dieser Arbeit vorgestellt wird, entstanden. Dieses Konzept wird an mehreren Beispielen veranschaulicht und auf die analysierten Prozesse angewendet. Diese Modelle werden ebenfalls beschrieben, wobei hier besonders auf die Änderungen gegenüber der vorherigen Modellierung eingegangen wird. Zudem wird argumentiert, weshalb die entsprechenden Konzepte verwendet wurden und ob es gegebenenfalls Alternativen gibt. Auch auf die Besonderheiten der einzelnen Modelle wird speziell eingegangen, um diese nochmals hervorzuheben.

Die Prozessmodelle, sowie die vorgeschlagene Erweiterung müssen noch gegen die in Tabel-

4 Methodik und Vorgehensweise

le 3.1 genannten Anforderungen validiert werden. Hierfür werden die einzelnen Anforderungen nochmals betrachtet und geprüft, ob sie erfüllt wurden.

Erweiterung der BPMN

In diesem Kapitel wird die Notwendigkeit einer Notation zur Darstellung von Kontextabhängigkeit anhand von Beispielen aus zwei unterschiedlichen Domänen aufgezeigt. Bei letzteren handelt es sich zum einen um die Medizin und zum anderen um das Software Engineering. Zwei Domänen, welche auf den ersten Blick keinerlei Gemeinsamkeiten haben. Dass diese jedoch vorhanden sind, wird am Ende des Kapitels aufgezeigt. Ein wichtiger Teil dieses Kapitels ist die Betrachtung und Diskussion alternativer Möglichkeiten der Modellierung von Kontext mit BPMN eigenen Mitteln. Im Anschluss an die Betrachtung der Prozesse innerhalb der Domänen und der Alternativen wird ein Vorschlag für neue BPMN-Symbole aufgezeigt und erläutert. Diese dienen der Modellierung von Kontexteinflüssen auf Prozesse. Um die Praxistauglichkeit der Symbole aufzuzeigen, werden die Prozessmodelle aus den Beispielen mit den neuen Symbolen nochmals modelliert. Hierbei wird auch ein weiteres Mal auf die Kontextabhängigkeiten der einzelnen Modelle eingegangen. Besonderer Augenmerk liegt bei dieser Betrachtung auf der Lösung eventueller Probleme durch den Vorschlag.

5.1 Prozesse in der Medizin

Die Anwendungsdomäne Medizin umfasst eine Vielzahl von Prozessen aus den unterschiedlichsten Subdomänen. Die meisten von ihnen sind direkt an die wichtigste Ressource der Domäne gebunden - den Patienten. Dieser ist Dreh- und Angelpunkt, wichtigster kontextueller und teils auch limitierender Faktor dieser Prozesse. Beispiele für Subdomänen, welche direkt mit dem Patienten interagieren, sind die Chirurgie, die hausärztliche Versorgung und auch Teilbereiche der Pathologie. Subdomänen, welche keinen direkten Zugang zu dem Patienten haben, wären unter anderem die Krankenhauslogistik, die Verwaltung oder die Labormedizin. In der vorliegenden Arbeit wurde die Subdomäne Anästhesie gewählt.

Der, bereits erwähnte, Faktor Patient ändert sich nicht nur von Instanz zu Instanz, sondern auch im Laufe dieser. Gerade bei invasiven Therapie- oder Untersuchungsformen kann dieser Faktor erheblichen Einfluss auf den Prozessverlauf nehmen. Er ist jedoch nicht der einzige Faktor,

welcher betrachtet werden muss. Auch der Arzt kann ein solcher Faktor sein. Je nach seiner Erfahrung und seinen Kenntnisstand verändert sich seine Vorgehensweise und damit unter Umständen auch der Kontext.

5.1.1 Modellierung ohne Kontextabhängigkeit

Im Folgenden werden Prozesse aus dem Bereich Medizin betrachtet. Hierfür werden die Prozesse in BPMN modelliert und textuell beschrieben. Im Anschluss erfolgt eine Analyse der Prozesse auf Kontextsensibilität und eventuelle Engstellen. Aus Gründen der Übersichtlichkeit und zum besseren Verständnis, sind die kontextabhängigen Prozessschritte rot ausgefüllt und mit einem unbeschrifteten Datenelement versehen. Dieses soll den Kontexteinfluss symbolisieren, Resultate, welche aus sich veränderndem Kontext hervorgehen, wurden, sofern nicht explizit erwähnt, nicht in den modellierten Prozess aufgenommen.

Narkoseeinleitung (Prä-operative Phase)

Der hier analysierte Prozess spiegelt die Vorgehensweise der Narkoseeinleitung bei der neurochirurgischen Anästhesie wider. Zur Verbesserung der Übersichtlichkeit wurde der Prozessgraph in mehrere Teile aufgesplittet, die einzelnen Prozessteile sind in den Abbildungen 5.1 bis 5.7 zu finden.

Die Narkoseeinleitung kann in drei Phasen gegliedert werden, wobei jede Phase Arbeitsschritte oder Orte zusammenfasst. In den zuvor genannten Abbildungen sind diese Phasen farbig voneinander getrennt (Phase 1 gelb, Phase 2 orange, Phase 3 rot). Die textuelle Beschreibung des Prozesses wird ebenfalls in die bereits erwähnten Phasen gegliedert.

- Phase 1

Die erste Phase der Narkoseeinleitung umfasst Arbeitsschritte, um den Patienten auf die Narkose vorzubereiten und markiert zudem den Beginn des Prozesses. Dieser startet damit, dass der Anästhesist den Einleitungsraum betritt. Er erkundigt sich nach dem Wohlbefinden des Patienten und vergleicht in dieser Zeit die Patientendaten. Diese Schritte sind nötig, um eine eventuelle Verwechslung ausschließen zu können. Da der Patientename allein durchaus mehrmals vorkommen kann, wird zur bereits erwähnten Absicherung das Geburtsdatum des Patienten mit der Akte verglichen. Da es vorkommen kann, dass die Angaben in der Akte unvollständig oder nicht zu 100% aktuell sein können, wird an dieser Stelle zudem nach bekannten Allergien gefragt und ebenfalls mit der Akte abgeglichen. Zudem ist es wichtig, den Zahnstatus des Patienten zu überprüfen. Sollte der Patient über Zahnprothesen verfügen, so müssen diese nun eingesetzt werden. Als letzte Absicherung wird der Patient nochmals gefragt, ob er nüchtern ist. Die zuvor beschriebenen Schritte erfolgen parallel zu einem kleinen Small Talk, um den Patienten zu beruhigen.

Nachdem die Identität des Patienten gesichert wurde, kann der Patient auf die Aufnahme

und Überwachung seiner Vitalparameter vorbereitet werden. Im Konkreten bedeutet dies das Anbringen von selbstklebenden EKG-Elektroden, das Anlegen einer Blutdruckmanschette und eines Fingerclips zur Überwachung der Sauerstoffsättigung im Blut. Diese Schritte bilden zudem den Abschluss der ersten Phase. Die erste Gruppierung zu finden in Abbildung 5.1 zeigt den beschriebenen Ablauf. Diese Phase orientiert sich an der WHO Surgical Safety Checklist. Siehe hierzu [WHO].

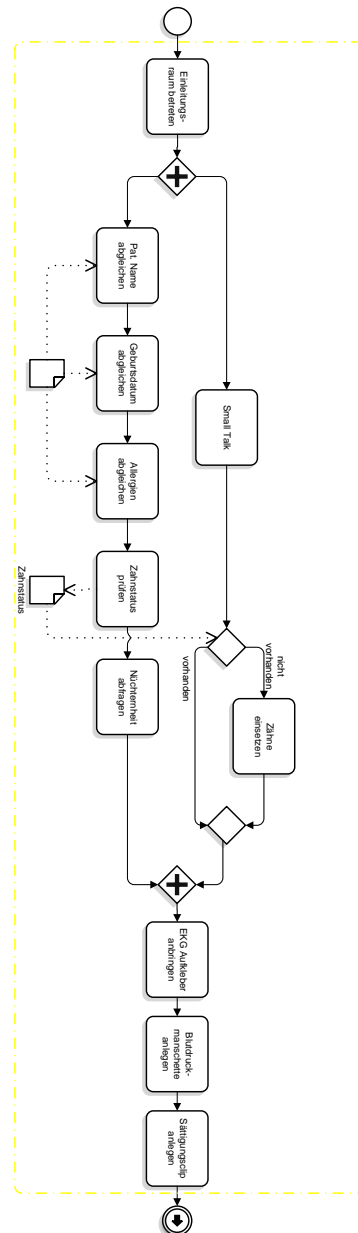


Abbildung 5.1: Narkoseeinleitung (Teil 1 von 6)

- Phase 2

Die zweite Phase des Narkoseeinleitungsprozesses ist, ausgehend von der Anzahl der

Arbeitsschritte, die längste der drei Phasen und wurde deswegen auf drei Abbildungen aufgeteilt, welche jeweils zu Beginn der Analyse referenziert werden. Die nachfolgenden Schritte sind in Abbildung 5.3 zu finden.

Nachdem in der ersten Phase die Identität des Patienten geprüft wurde und der Patient auf die Überwachung seiner Vitalparameter vorbereitet wurde, wird der erste Zugang gelegt und gesichert. Dieser wird benötigt um dem Patienten Medikamente intravenös zu verabreichen. Wenn der Zugang liegt, wird dem Patienten eine Atemmaske auf Mund und Nase gehalten und der Patient wird angewiesen mehrmals tief ein- und auszuatmen. Die Maske liefert ein Gemisch aus Sauerstoff und Inhalationsanästhetikum. Der nächste Schritt umfasst einen eigenen Subprozess und beschreibt die eigentliche Narkoseeinleitung. Dazu werden dem Patienten über den zuvor gelegten Zugang drei Medikamente gegeben. Als erstes wird ein Opiat intravenös verabreicht. Dieses stellt das Schmerzempfinden des Patienten ab und gewährleistet, dass dieser während der OP nicht auf Grund von Schmerz aufwacht. Das zweite ist ein Hypnotikum, welches den Patienten binnen Sekunden einschlafen lässt. Zuletzt erhält der Patient ein Muskelrelaxans. Dieses entspannt die Muskulatur des Patienten, wodurch der darauffolgende Schritt durchgeführt werden kann, welcher wiederum ein Subprozess ist.

Die endotracheale Intubation, zu finden auf den Abbildungen 5.4 und 5.5, beginnt damit, dass der Patient in die sogenannte Jackson-Position gebracht wird. Hierfür wird der Kopf des Patienten hochgelegt und sein Nacken leicht überstreckt. Daraufhin wird der Kiefer des Patienten geöffnet und das Laryngoskop (siehe [LAR]) eingeführt. Mittels diesem wird die Epiglottis (Kehledeckel) gehoben, dies ermöglicht den Anästhesisten die Speiseröhre und die Trachea (Luftröhre) anzusehen und zu entscheiden, ob er eine Einführhilfe verwenden möchte oder nicht. Bei angehobener Epiglottis kann nun die Magensonde eingeführt werden, über welche überflüssiger Magensaft abgeführt wird. Sobald die Magensonde gelegt ist, wird der Endotrachealtubus (Abbildung 5.2) eingeführt. Liegt dieser in der gewünschten Position kann der, an der Tubusspitze angebrachte, Cuff mittels einer Spritze aufgeblasen werden und der Cuffdruck mit Hilfe des Cuffdruckmessers geprüft und angepasst werden. Diese Aktion ist die letzte im Subprozess endotracheale Intubation. Im Anschluss daran werden der Tubus und die Magensonde mit Leukoplast fixiert. Zum Schutz der Augen des Patienten wird ein Augengel auf die Netzhaut aufgetragen, bevor die Augen abgeklebt werden. Falls die OP-Dauer zwei Stunden überschreitet, wird an dieser Stelle ein Blasen-katheter gelegt. Ab hier beginnt eine engmaschige Kontrolle der Vitalparameter des Patienten. Sobald der OP-Saal bereit ist, werden die Blutdruckmanschette, das EKG und die Beatmung vom Patienten getrennt und der Patient wird aus dem Einleitungsraum hinaus geschoben.

- Phase 3

Den Beginn der dritten Phase markiert der Transport des Patienten in den OP-Saal, wo als erstes der Patient an ein Narkosegerät angeschlossen wird. Dieses Gerät dient der Überwachung der Vitalparameter und der Beatmung des Patienten mittels mit Narkosegas



Abbildung 5.2: Endotrachealtubus vom Typ Magill-Tubus

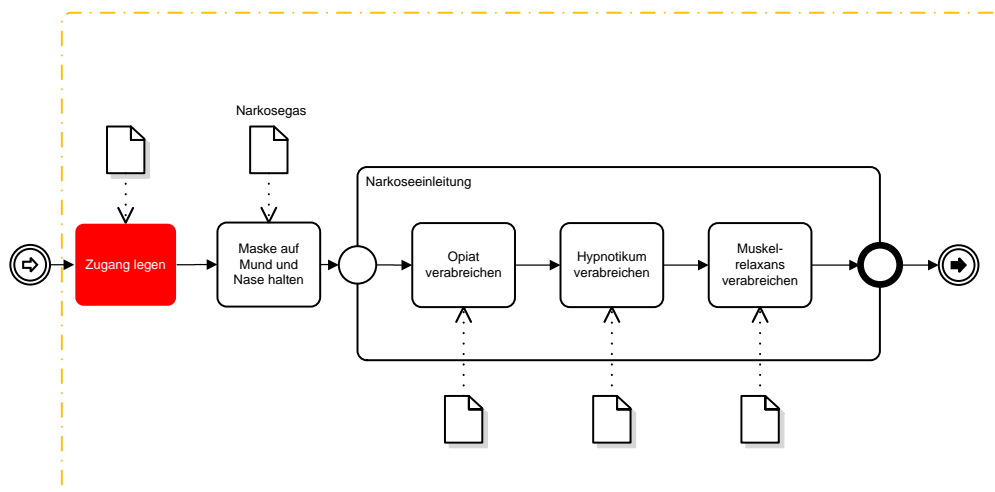


Abbildung 5.3: Narkoseeinleitung (Teil 2 von 6)

angereichertem Sauerstoff. Als nächstes wird der Patient an ein Blutdruckmessgerät sowie an die benötigte Medikation angeschlossen. Um alle Vitalparameter erfassen zu können, wird der Patient nun noch an die bereits vorhandenen EKG-Aufkleber angeschlossen und es wird ein Clip zu Blutgasmessung angelegt. Nun ist es möglich die Vitalparameter des Patienten abzulesen und zu dokumentieren. Ein Punkt, welcher ebenfalls dokumentiert werden muss, ist die Narkose an sich. Das bedeutet konkret die Dokumentation der Menge sowie Bezeichnung der verwendeten Medikamente. Dieser Schritt umfasst zudem die Dokumentation des OP-Teams und der Narkoseeinleitungszeiten. Sobald alle benötigten Daten dokumentiert wurden, unterstützt das Anästhesieteam den Rest des OP-Teams bei der Lagerung des Patienten, was soviel bedeutet wie den Patienten in eine Position zu bringen und, sofern nötig, zu fixieren. Die Lage des Patienten ist hierbei abhängig von der OP und der Position der Operateure. Sobald der Patient in die festgelegte Position gebracht wurde, können die Stellen, an welchen ein Schnitt gesetzt werden soll, steril abgewaschen werden. Im Anschluss daran wird der Rest des Patienten steril abgedeckt. Sofern die OP länger als zwei Stunden dauert, wird der Patient zusätzlich noch mit einer Decke zugeeckt, welche mit warmer Luft gefüllt wird und den Patienten wärmt. Der letzte Schritt der

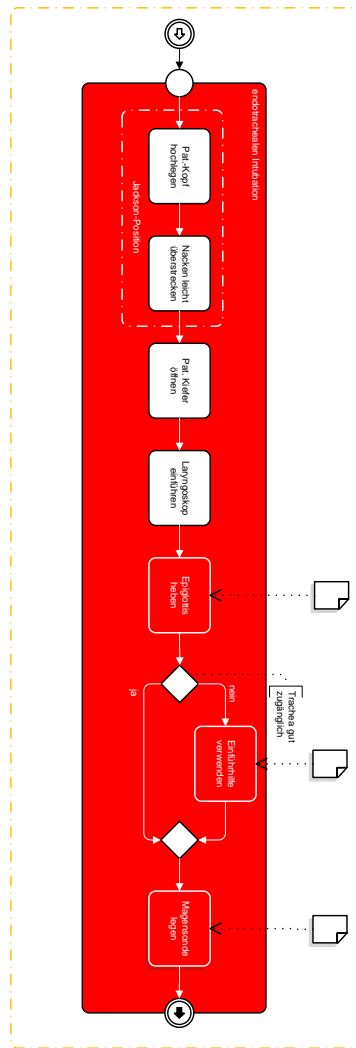


Abbildung 5.4: Narkoseeinleitung (Teil 3 von 6)

Narkoseeinleitung, beziehungsweise der präoperativen Phase, ist die Dokumentation der Uhrzeit des ersten Schnittes.

Die Prozessschritte der Phase 3 sind in den Abbildungen 5.6, 5.7 modelliert und durch die rote Gruppierung zusammengefasst.

- Allgemeines

Bei der Modellierung und Beschreibung des Prozesses wurde nicht darauf eingegangen, dass ein Anästhesie Team nicht nur aus einer Person besteht. Dieser Aspekt wurde herausgenommen, da die zusätzlichen Personen lediglich die Dauer des Prozesses reduzieren und keine bis kaum Auswirkungen auf den Prozessablauf haben. Personen deren Anwesenheit und Mitarbeit den Prozess verändern kann, werden im hierauf folgenden Analyseabschnitt betrachtet.

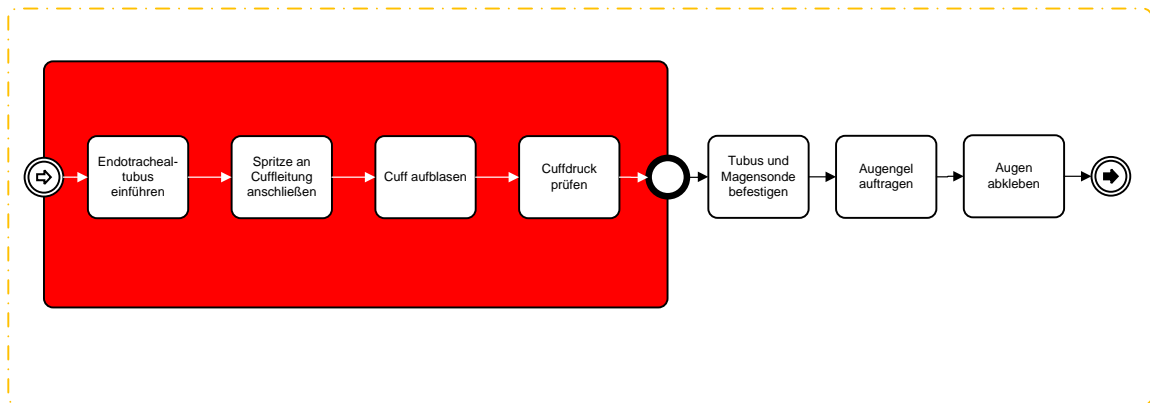


Abbildung 5.5: Narkoseeinleitung (Teil 4 von 6)

Nachdem nun ein Überblick über den Prozess geschaffen wurde, ist es möglich diesen zu analysieren. Wie eingangs bereits erwähnt sind die Aktivitäten, die besonders kontextsensitiv sind, rot hinterlegt. Zudem werden in diesem Abschnitt Aspekte des Prozesses mitaufgenommen, welche nicht modellierbar waren ohne den Prozess unnötig zu verkomplizieren (siehe hierzu 5.3). Die Analyse wird wie bereits die Beschreibung und die Modellierung in drei Phasen unterteilt.

- Phase 1

Die erste Phase des Prozesses erscheint auf den ersten Blick sehr geradlinig und frei von kontextuellen Einflüssen und wurde auch entsprechend modelliert. Dieser Eindruck täuscht jedoch. Genau wie der Rest des Prozesses hängt auch diese Phase stark von Kontext ab, wenn auch hier die Modellierung zum größten Teil gut möglich ist. Die Prozessschritte, auf welche hier besonders eingegangen wird, sind in Abbildung 5.8 nochmals aufgezeigt.

Betrachtet man den Prozess nochmals etwas genauer, erkennt man, dass bei den Prozessschritten *Pat. Name abgleichen*, *Geburtsdatum abgleichen* und *Allergien abgleichen* keine Aussage darüber getroffen wird, was geschehen soll, falls einer der abzugleichenden Daten nicht mit der Akte übereinstimmt. Es ist eindeutig, dass der Prozess nicht weitergeführt werden darf, solange der Missstand nicht aufgeklärt wurde. Doch muss der Prozess gleich komplett abgebrochen werden? Die Möglichkeiten für die Differenz sind vielfältig und hängen vom Schritt ab, in dem der Missstand auftritt. Falls der Patient einen häufig vorkommenden Namen hat (z. B. Peter Müller) oder es mehrere phonetisch identische Schreibweisen für den Namen gibt (Maier oder Meier oder Mayer) ist eine Verwechslung der Akte möglich. In diesem Fall würde es ausreichen den Prozess zu stoppen und die Akte gegen die korrekte zu tauschen. Die Fehlerquellen hierfür sind mannigfaltig. Zum Einen kann es an einem vollen Tag durchaus vorkommen, dass bei zwei Patienten gleichen Namens eine Verwechslung auftritt und zum anderen ist es gerade bei phonetisch gleichen Namen möglich, dass bereits bei der Aufnahme des Patienten ein Fehler geschehen ist,

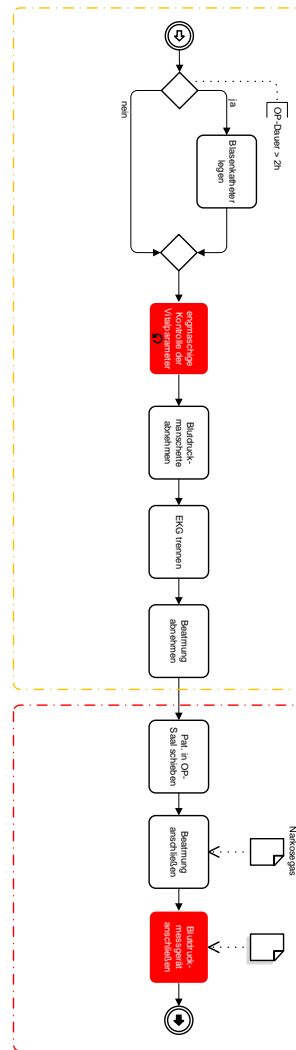


Abbildung 5.6: Narkoseeinleitung (Teil 5 von 6)

welcher sich erst jetzt durch eine Verwechslung bemerkbar macht.

Die Wahrscheinlichkeit zwei Patienten gleichen Namens zu haben ist durchaus hoch, dass beide Patienten am selben Tag geboren wurden ist dagegen relativ unwahrscheinlich. Daher ist auch der zweite Abgleich mit der Patientenakte ein kontextsensitiver Schritt. Hierfür gilt die selbe Vorgehensweise wie beim Schritt zuvor. Ebenso wichtig und kontextabhängig ist die letzte Aktivität der Identitätskontrolle, der Abgleich der Allergien. Hierbei ist besonders bei älteren Patienten Aufmerksamkeit höchstes Gebot, da sich Allergien erst später entwickeln können und die Patientenakte eventuell diesbezüglich nicht aktualisiert wurde. Sollte der Patient eine Allergie gegen ein zu verwendendes Medikament haben, kann dies unter Umständen lebensgefährlich werden. All diese Abhängigkeiten sind im Prozessgraph modelliert, was jedoch immer noch fehlt sind die Kompensationsschritte. Gründe hierfür sind im Kapitel 5.3 aufgeführt und werden dort ausführlich diskutiert.

Nicht alle kontextuellen Einflüsse sind gar nicht oder nur schwer modellierbar. Ein Beispiel,

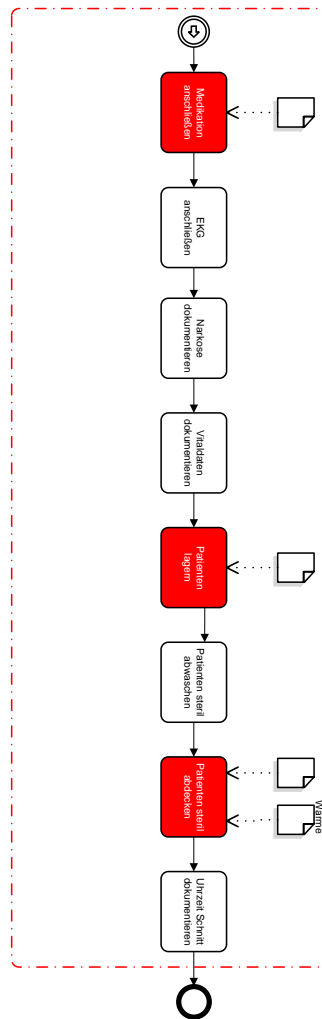


Abbildung 5.7: Narkoseeinleitung (Teil 6 von 6)

bei welchen es sehr gut funktioniert, ist ebenfalls in Abbildung 5.8 zu finden, die Aktivität *Zahnstatus prüfen*. Hier sieht man direkt die Auswirkung von Kontext auf den Prozessablauf. Die Aktivität *Zahnstatus prüfen* erzeugt und schreibt das Datenelement Zahnstatus. Dieses Element ist zugleich auch der lokale Kontext für die Aktivität (vergleiche mit Definition 2.2). Die Auswirkung zeigt sich an dem parallel abgefragten exklusiven Gateway, sollte der Patient keine Zahnprothese tragen (Zahnstatus = eigene Zähne) so müssen diese nicht eingesetzt werden. Selbiges gilt falls der Patient die Prothese zuvor selbstständig eingesetzt hat (Zahnstatus = Prothese eingesetzt) oder diese nicht vorhanden ist und aus diesem Grund nicht eingesetzt werden kann (Zahnstatus = Prothese nicht vorhanden). Sollte keiner der zuvor beschriebenen Zustände zutreffen, kann die Aktivität *Zähne einsetzen* ausgeführt werden.

Die restlichen Aktivitäten dieser Phase haben keine nennenswerten kontextuellen Abhängigkeiten. Sie werden erst interessant wenn der Prozess mit maximaler Granularität dar-

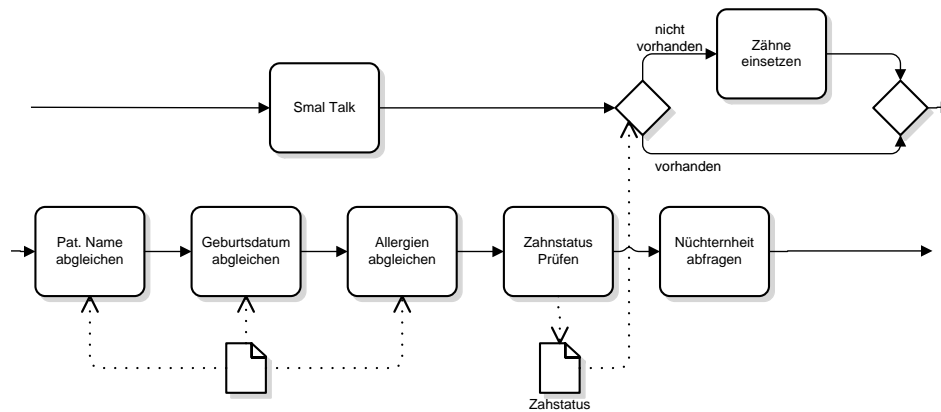


Abbildung 5.8: Narkoseeinleitung (Teil 1 von 6) Ausschnitt 1

gestellt werden soll. In einem solchen Fall wäre es ratsam, diese Bereiche des Prozesses textuell zu beschreiben. Ein weiterer Kontext, welcher hier von Interesse sein könnte, ist die vorhandene Materialart. Dies muss jedoch nicht explizit modelliert werden, da dieser Kontext durch den Anwender des Prozesses implizit gesetzt und unter Umständen verändert wird.

- Phase 2

Die zweite Phase beginnt direkt mit einer kontextabhängigen Aktivität, dem Legen eines Zugangs. Klassischerweise wird der Zugang am Handrücken gelegt, wie in Abbildung 5.9 exemplarisch dargestellt. Die gewünschte Vene ist allerdings nicht immer zugänglich. Gründe hierfür können Tätowierungen, sogenannte Rollvenen oder allgemein schlecht zugängliche Venen sein. Auch ist es möglich, dass beim Einführen der Kanüle ein Nerv getroffen wird, weswegen ein neuer Zugang gelegt werden muss. Zudem müssen in bestimmten Fällen weitere Zugänge gelegt werden. Beispiele für diese Fälle sind eine lange OP-Dauer oder die Lagerung des Patienten während der OP. Diese Umstände und vor allem die Kombination aus diesen sind jedoch nur schwer bis gar nicht modellierbar. Nachdem die Zugänge gelegt worden sind, folgen zuerst einige kontextuell wenig bis gar nicht beeinflusste Aktivitäten. Die aus kontextueller Sicht interessanteste von ihnen ist der Subprozess der Narkoseeinleitung. Hier kann es, trotz zuvor erfolgreichem Abgleich, zu einer Unverträglichkeitsreaktion des Patienten kommen.

Einen Gegensatz zum bereits beschriebenen stellt der Subprozess *endotracheale Intubation* dar. Dieser ist stark kontextabhängig, was sich vor allem in den darin enthaltenen Aktivitäten widerspiegelt. Betrachtet man diese, zeigt sich zu Beginn ein eher weniger kontextabhängiges Bild. Das Bringen des Patienten in die Jackson-Position ist nicht von äußeren Faktoren beeinträchtigt. Dies würde bei einer Notoperation eventuell anders aussehen, da bei einem Patienten mit Verletzungen im Hals-Nacken-Bereich das Überdehnen

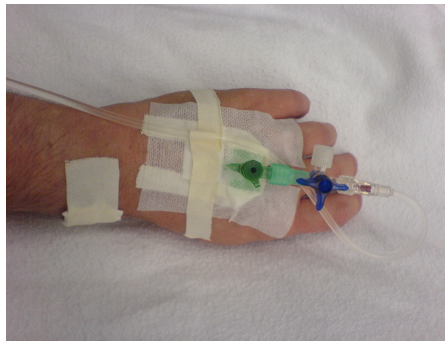


Abbildung 5.9: Intravenöser Zugang zu einer Handvene, mit Dreiweghahn (Quelle: commons.wikimedia.org)

des selbigen zu weiteren Schädigungen führen könnte. Der hier betrachtete Prozess geht jedoch von einer planmäßigen Operation ohne solche Beeinträchtigungen des Patienten aus. Auch das Öffnen des Patientenkiefers ist, mit ein wenig Übung, bei einer narkotisierten Person gut durchführbar. Interessanter wird es beim Einführen des Laryngoskops. Zum Einen muss hier vom Anästhesisten die passende Größe gewählt werden und zum Anderen wird der Kontext für die darauf folgenden Schritte erst jetzt erkennbar und auswertbar. Ein Umstand, welcher bei klassischer Modellierung Probleme bereitet. Unter Zuhilfenahme des zuvor genannten Werkzeugs kann der Arzt nun die Epiglottis heben, um die Veränderungen des lokalen Kontext zu betrachten. Dieser ist die Trachea des Patienten und die Zugänglichkeit dieser. Diese Information wird verwendet um eine Entscheidung bezüglich der Verwendung einer Einführhilfe zu treffen. Der visuelle Eindruck der Epiglottis mag den Arzt zu der Entscheidung führen keine Einführhilfe zu verwenden, führt beim Versuch die Magensonde einzuführen allerdings zu Problemen, weil eine eventuelle Verengung nicht sichtbar war. Auch die Art der Einführhilfe hängt von den Vorlieben des Arztes und der gegebenen Möglichkeiten ab. Die Darstellung aller möglicher Kombinationen ist weder empfehlenswert noch hilfreich, Begründungen hierfür sind im Kapitel 5.3.2 aufgeführt.

Der nächste Schritt mit hohem Kontextanteil ist das Legen der Magensonde. Im optimalen Fall lässt sich diese am Laryngoskop entlang, die Speiseröhre herab bis zum Magen führen. Nun kann es allerdings passieren, dass sich die Sonde im Rachenraum oder in der Speiseröhre verkanntet und sich so ein "Knoten" bildet. Um die Verletzungsgefahr des Patienten zu verringern, greift der Arzt in solchen Fällen zu einer sogenannten Magill-Zange. Damit lässt sich die Sonde greifen und vorsichtig entnehmen. Zudem lässt sich mittels der bereits genannten Zange die Sonde relativ tief in die Speiseröhre einführen. Liegt die Magensonde in der gewünschten Lage, lässt sich unter Beachtung selbiger Problematik der endotracheale Tubus einführen. Auch hier muss die Entscheidung über die Verwendung und Art der Einführhilfe vom Arzt getroffen werden. Der Tubus wird in die Luftröhre gelegt und dort fixiert. Das Fixieren der zuvor gelegten Hilfsmittel am Patient ist nur soweit kontextabhängig, dass die Beschaffenheit und Form der Gesichts von Patient zu Patient abweicht. Ebenso verhält es sich für die nachfolgenden Schritte, dem Auftragen von Augengel und dem Abkleben der Augen. Interessanter wird es in den nächsten Schritten. Wie

bereits zuvor erwähnt, ist es möglich, dass bei einer OP-Dauer länger als zwei Stunden Aktivitäten durchgeführt werden müssen, welche ansonsten nicht nötig gewesen wären. Eine dieser Aktivitäten ist das Legen eines Blasenkatheters. Die Aktivität markiert auch den Abschluss der vorbereitenden Tätigkeiten und den Beginn einer kontextuell höchst brisanten Aktivität, der *engmaschigen Kontrolle der Vitalparameter*. Dieser Prozessschritt ist besonders bei kritischen Patienten wichtig und erstreckt sich über den Rest des Prozesses und in die darauf folgenden Prozesse mit hinein. Sollten die Vitalparameter sich zum Nachteil des Patienten ändern, so muss sofort reagiert werden. Dadurch ist es möglich, dass die gesamte Prozessausführung sich ändert oder dass ad-hoc-Aktivitäten temporär eingefügt werden. Die restlichen Aktivitäten dieser Phase unterliegen nur soweit kontextuellem Einfluss, dass sie mit den korrespondierenden Aktivitäten zu Beginn dieser Phase zusammenhängen.

- Phase 3

Wie bereits beschrieben, beginnt diese Phase mit einem Ortswechsel. Der Patient wird vom Einleitungsraum in den OP-Saal gebracht und wie gehabt an die mit Narkosegas angereicherte Beatmung angeschlossen. Der kontextuell interessante Arbeitsschritt folgt auf das bereits Geschehene. Nun muss der Patient an ein Blutdruckmessgerät angeschlossen werden, die Art der Blutdruckmessung hängt nun wiederum von unterschiedlichen Faktoren ab. Einer dieser Faktoren ist die Lagerung des Patienten. Nachdem der Patient an die Blutdruckmessung angeschlossen wurde, muss die Medikation bestimmt und ebenfalls verbunden werden. Hier gilt wiederum Menge, Art und Verabreichungsgeschwindigkeit hängen von der OP-Art und -Dauer ab, weswegen die Modellierung dieser nicht empfehlenswert ist (vergleiche hierzu Kapitel 5.3.2). Die erste näher betrachtete Aktivität in dieser Phase, ist die Lagerung des Patienten. Der Kontext dieser ergibt sich wiederum aus mehreren Quellen. Zum Einen ist die zu operierende Stelle ausschlaggebend und zum anderen der Operateur und die Position, an welcher er stehen möchte, um die OP möglichst gut durchführen zu können. Es muss jedoch auch betrachtet werden, ob weitere Schnitte gesetzt werden müssen. Auch ist bei einer OP meist nicht nur ein Operateur zugegen, daher müssen die Positionen dieser Operateure, wenn möglich, ebenfalls mit in die Betrachtung einfließen. Wenn der Patient in entsprechender Position liegt und steril abgewaschen wurde, erfolgt der nächste kontextabhängige Schritt: Das sterile Abdecken des Patienten. Auch hier muss abermals Rücksprache mit den Operateuren gehalten werden, um ihnen die Arbeit nicht unnötig zu erschweren. Zudem wirkt sich auch hier die OP-Dauer aus. Sollte die OP länger als zwei Stunden dauern, muss der Patient mit einer Wärmedecke zugedeckt werden. Dieser Umstand ist jedoch zu OP-Beginn unter Umständen nicht ersichtlich. In einem solchen Fall muss dies während der Operation geschehen. Immer vorausgesetzt, dass die Lagerung des Patienten und der zu operierende Bereich dies zulassen.

Intraoperative Phase und postoperative Phase

Als zweiter medizinischer Prozess wurden die logischen Folgeprozesse des zuvor betrachteten Narkoseeinleitung Prozesses gewählt. Es handelt sich hierbei um die Prozesse der intraoperativen Phase sowie der postoperativen Phase. Die intraoperative Phase beschreibt die Zeit zwischen Hautschnitt und Naht und die postoperative Phase die Zeit zwischen Naht und Verlegung auf Station. Wie beim zuvor beschriebenen Prozess wurde auch hier die Anästhesie-Seite betrachtet.

Der nachfolgende Prozess ist in Abbildung 5.10 in BPMN modelliert. Es handelt sich dabei um die Intraoperative Phase. Die Hauptaufgabe der Anästhesie während einer laufenden OP umfasst vor allem die Kontrolle der Vitalparameter des Patienten und der Narkose. Es bestehen jedoch auch zusätzliche Dokumentationsschritte, welche hier erledigt werden müssen. Falls bereits entschieden wurde, ob der Patient nach der OP auf die Intensivstation verlegt werden soll, wird vom Anästhesisten ein Plan für die Weiterbehandlung des Patienten seitens der Intensivmedizin geschrieben. Sofern der Patient nach der OP auf eine Normalstation gebracht wird, entfällt dieser Schritt und der Arzt fährt mit der Bearbeitung der nächsten Aktivität fort, der Dokumentation des OP-Teams. Das Ende dieser Phase wird durch das Setzen der Naht markiert, auch dieser Zeitpunkt muss auf dem OP-Protokoll notiert werden.

Da der zuvor beschriebene Prozess verhältnismäßig kurz ausfällt, wurde die nachfolgende OP Phase ebenfalls in die Betrachtung miteinbezogen. Es handelt sich dabei um die postoperative Phase. Sie ist in den Abbildungen 5.11 und 5.12 zu finden. Nachdem die Chirurgie ihre Arbeit abgeschlossen hat, kann der Patient für den Aufwachraum vorbereitet werden. Hierfür werden zunächst die sterilen Decken entfernt, um ihn anschließend von der Beatmung trennen zu können. Auch kann der Patient vom EKG getrennt werden. Sollte er eine Blutdruckmanschette getragen haben, wird diese nun abgenommen, ebenso der zuvor angebrachte Schutz der Augen. Sobald der Patient von den Geräten getrennt wurde, kann er auf ein Bett verlagert und in den Aufwachraum gebracht werden, wo er an bereits erwähnte Messgeräte angeschlossen wird.

Hier wird zunächst der OP-Bericht eingescannt und abgespeichert. In dieser Zeit sollte auch die Wirkung des Hypnotikums nachlassen und der Patient langsam erwachen. In diesem Fall kann nun die Magensonde gezogen und der Tubus entfernt werden, da der Patient nun auch eigenständig atmen kann. Zur Unterstützung wird ihm nun einige Minuten reiner Sauerstoff gegeben. Während der gesamten Prozedur müssen auch hier die Vitalparameter überwacht werden. Sobald der Patient ansprechbar ist, wird dieser nach seinem Befinden gefragt. Falls der er keine Beschwerden hat wird er über den OP Verlauf informiert. Sobald der Patient stabil, ist kann er auf die zuvor festgelegte Station verlegt werden.

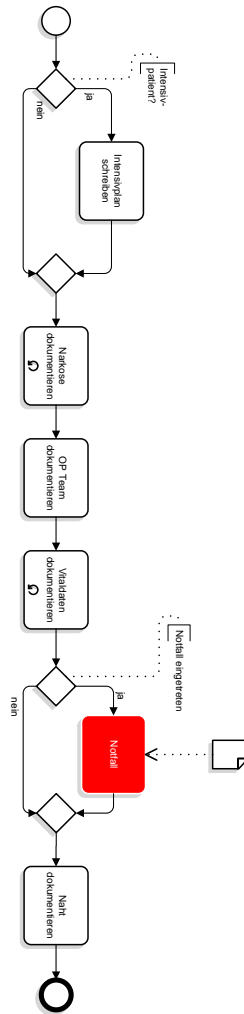


Abbildung 5.10: intraoperative Phase

Analyse

Auch diese Prozesse werden nun auf ihre kontextuellen Engpässe untersucht. Diese werden an der Position des Auftretens beschrieben.

Der Prozess der intraoperativen Phase wurde im vorliegenden Beispiel mit sequenziellem Verlauf dargestellt, diese Darstellung wurde aus Gründen der höheren Übersichtlichkeit und damit bessere Erklärbarkeit gewählt. Man kann sich durchaus andere Anordnungen der einzelnen Aktivitäten vorstellen. Vor allem die Aktivität *Narkose dokumentieren* ist immer zu dem Zeitpunkt wichtig, in welchem dem Patienten weitere Narkosemittel zugeführt werden. Ähnlich verhält es sich mit den Vitalparametern des Patienten, diese müssen in festgelegten Intervallen dokumentiert werden. Dies bedeutet jedoch nicht, dass diese auch nur zu diesen Zeitpunkten abgelesen werden. Sie müssen während der gesamten OP stets überwacht werden, um einen nahenden Notfall unter Umständen frühzeitig zu erkennen und entsprechend reagieren zu können. Tritt



Abbildung 5.11: postoperative Phase (1 von 2)

trotz der engmaschigen Kontrolle ein Notfall ein, so muss auf diesen schnell reagiert werden. Ein Notfall bedeutet eine drastische Änderung des Kontext, was im schlimmsten Fall zur kompletten Verwerfung des laufenden Prozesses, sprich der laufenden und auch der nachfolgenden Phasen führen kann.

Ein vergleichbares Verhalten zeigt auch der zweite Prozess. Auch hier ist der kritische Schritt direkt mit den Vitalparameter des Patienten verbunden. Sollten diese sich ändern und dadurch vorbestimmte Grenzwerte über- oder unterschreiten, so muss unverzüglich darauf reagiert und der Patient stabilisiert werden. Ebenso sollte der Arzt die Antwort des Patienten berücksichtigen. Wenn der Patient über Übelkeit klagt, so kann es sich um eine Unverträglichkeitsreaktion auf die Narkose handeln. Auf diese muss entsprechend reagiert werden, indem dem Patienten eine Nierenschale bereitgestellt wird, falls er sich übergeben muss, um ein Beispiel zu nennen.

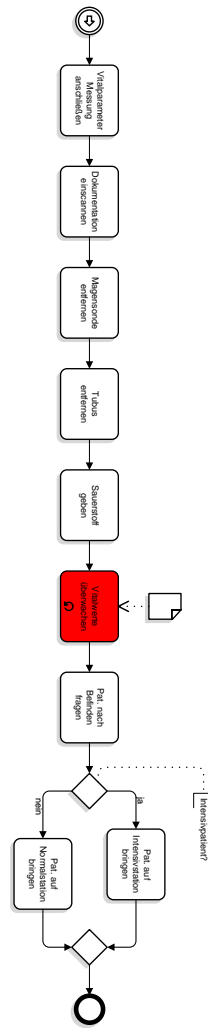


Abbildung 5.12: postoperative Phase (2 von 2)

Die zuvor beschriebenen und analysierten Prozesse stehen exemplarisch und stellvertretend für die Anwendungsdomäne Medizin. Sie bieten einen kleinen Einblick in diese höchst komplexe Domäne.

5.2 Prozesse im Software Engineering

Wie auch bei der ersten Anwendungsdomäne umfasst diese mehrere Subdomänen, welche unterschiedlichen Zwecken dienen. Anders als in der Medizin müssen die Subdomänen hier zusammenarbeiten, um das Ziel - eine funktionierende Software - zu erreichen. Auch gibt es sinnvolle Abfolgen dieser. Diese Abfolgen sind die eigentlichen Software Engineering Prozesse. In der vorliegenden Arbeit wurde lediglich auf die Subdomäne Programmierung eingegangen. In dieser Domäne wird die Anwendung in einer Festgelegten Programmiersprache realisiert. Auch

diese Domäne verfügt über spezifische kontextuelle Einflüsse. Beispiele hierfür sind Bugfixes, Ad-hoc Aktivitäten, personeller Ausfall oder Probleme im Code. Ebenso muss auf diese Veränderungen eingegangen und reagiert werden.

Eine vollständige Umsetzung der hier diskutierten Prozessmodelle in BPMN 2.0 ist unter [Men14] nachlesen.

5.2.1 Modellierung ohne Kontextabhängigkeit

Wie auch schon in der vorangegangenen Anwendungsdomäne erfolgt auch hier nun die Beschreibung zweier ausgewählter Prozesse. Diese wird sowohl in BPMN als auch textuell erfolgen. Die Beschreibungen erfassen zum Einen die einzelnen Prozesselemente und zum Anderen Randinformationen zum Prozessmodell an sich. Diese dienen der Unterstützung des Prozessverständnisses und liefern einen kleinen Überblick über die Rahmenbedingungen des selbigen. Stellvertretend für die Anwendungsdomäne wurden in der vorliegenden Arbeit der *Open Unified Process* und der *Scrum*-Prozess gewählt.

Der Open Unified Process

Der *Open Unified Process*, ab dieser Stelle als OpenUP bezeichnet, ist ein von der Eclipse Foundation entwickelter Softwareentwicklungsprozess. OpenUP gehört zu den agilen Prozessen. Er besteht aus mehreren Phasen und ist iterativ und erweiterbar. Weitere Informationen zu OpenUP sind unter [Ecl] zu finden. In dieser Arbeit wird lediglich auf einen Ausschnitt des Gesamtprozesses eingegangen, welcher in Abbildung 5.13 abgebildet ist.

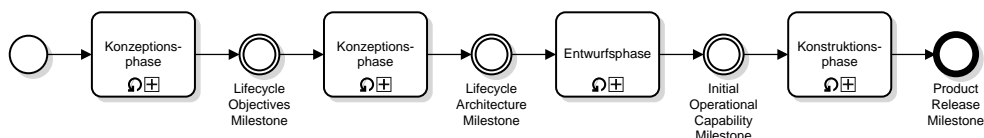


Abbildung 5.13: OpenUP Lifecycle, Quelle: [Men14]

Der Subprozess, welcher genauer betrachtet wird, ist in beinahe jeder Phase des Prozesses zu finden und wird als *Develop Solution Increment* Prozess bezeichnet. Dieser Prozess eignet sich für die kontextuelle Betrachtung aufgrund seiner, auf den ersten Blick, einfachen Struktur. Die nachfolgende textuelle Beschreibung bezieht sich in Abbildung 5.14. Die Arbeitsaufgaben, die diesem Prozess zugrunde liegen, können Anwendungsfälle, Szenarien, Support-Anfragen oder auch ein Änderungsauftrag sein.

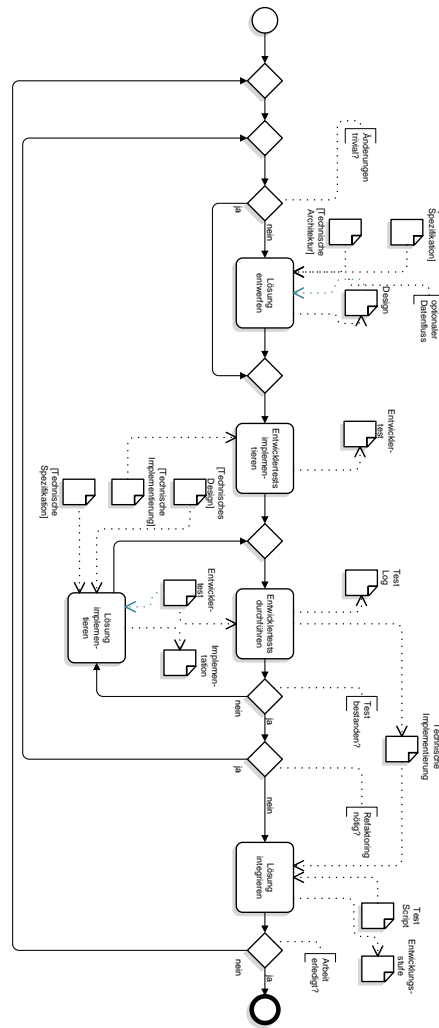


Abbildung 5.14: Develop Solution Increment Prozess, Quelle: [Men14] mit eigenen Anpassungen

Die folgende Beschreibung bezieht sich auf den Prozess, welcher in Abbildung 5.14 dargestellt ist.

Zu Beginn des Prozesses muss geprüft werden, ob die angeforderten Änderungen trivial sind. Diese Prüfung ermöglicht die Entscheidung über die Notwendigkeit eines Lösungsentwurfs. Bei trivialen Änderungen ist es nicht nötig einen solchen anzufertigen und man kann sofort mit der Entwicklung der Lösung beginnen. Falls die Änderungen nicht trivial sein sollten, muss ein Lösungsentwurf angefertigt werden. Hierfür werden die technischen Spezifikationen sowie die technische Architektur berücksichtigt werden. Ebenso verhält es sich mit eventuell bereits vorliegenden Lösungsdesigns. Die Betrachtung dieser ist jedoch optional. Steht das Design der Lösung fest können die Entwicklertest implementiert werden. Auch hier spielen die technische Architektur sowie unter Umständen das technische Design eine wichtige Rolle. Zusätzlich zu den Tests wird auch das Lösungsdesign an dieser Stelle implementiert. Nach erfolgreichem Im-

plementieren der Tests und der Lösung können diese durchgeführt werden. Besteht die Lösung den Test nicht, so muss sie neu implementiert und anschließend erneut getestet werden. Dies geschieht solange, bis die implementierte Lösung die Test besteht. Nach Abschluss dieser Tests kann es jedoch nötig sein, dass die Lösung refaktoriert werden muss. Dies soll die Lesbarkeit, Wartbarkeit und Erweiterbarkeit des Codes verbessern. Wichtig dabei ist, dass das Lösungsverhalten nicht verändert wird. Sollte die Entscheidung darauf fallen, dass die Lösung refaktoriert werden soll, beginnt der Prozess von neuem mit der Betrachtung der Änderungen. Fällt die Entscheidung gegen diese Maßnahme, wird die Lösung in das Gesamtsystem integriert. Sofern diese die letzte fehlende Komponente des Systems war, endet der Prozess an dieser Stelle. Sollte weitere Komponenten zu entwickeln sein, startet, wie bereits beim Refactoring, der Prozess von neuem.

Als nächster Schritt muss dieser Prozess nun auf seine kontextuellen Abhängigkeiten untersucht werden. Der Prozess hat nur wenig Aktivitäten, welche bei oberflächlicher Betrachtung alle kontextuell beeinflusst werden. Bereits beim Entwerfen der Lösung existieren mehrere Einflussfaktoren, deren Auswirkung berücksichtigt werden muss. Es ist jedoch nicht zwangsläufig nötig, die laufende Aktivität zu unterbrechen, um auf den geänderten Kontext zu reagieren. Ebenso verhält es sich bei der letzten Aktivität des Modells. Auch hier können sich noch kurzfristige Änderungen ergeben, welche eingepflegt werden müssen. Die Implementierung und Durchführung der Entwickler-Tests, sowie die Implementierung der eigentlichen Lösung wurden bei dieser Betrachtung herausgenommen. Der Grund hierfür ist einfach zu erläutern. Erlaubt man bei jedem Schritt kontextuelle Beeinflussung, ist es möglich, dass der Prozess in eine Endlosschleife gerät. In Kapitel 5.5 werden jedoch zwei alternative Modellierungen angegeben. Eine, die sich auf diese Analyse stützt und eine zweite, welche auf der anfangs aufgestellten Behauptung basiert, dass jede Aktivität kontextuell beeinflussbar ist.

Der Scrum-Prozess

Die Bezeichnung *Scrum* stammt aus dem Rugby und beschreibt dort einen Spielzug. Wörtlich übersetzt bedeutet es *Gedränge*. Im Software Engineering steht es für sehr bekannten agilen Softwareentwicklungsprozesse. Scrum hat einen einfachen und iterativen Aufbau, wodurch er sich leicht erlernen lässt. Er basiert auf dem Agilen Manifest, nachzulesen unter [BBB⁺01], welches die Gewichtung bei agiler Software Entwicklung auf folgende Punkte setzt:

- Menschen und Interaktionen vor Prozessen und Werkzeugen
- Funktionierende Software vor umfassender Dokumentation
- Zusammenarbeit mit Kunden vor Vertragsverhandlungen
- Reagieren auf Veränderungen vor dem Befolgen eines Plans

Drei der Autoren des zuvor genannten Manifestes haben Scrum in den 1990er-Jahren entwickelt und sind bis heute an seiner Weiterentwicklung beteiligt. Eine ausführliche Beschreibung von Scrum ist unter [NTM14] zu finden. Im Folgenden wird ausschließlich auf den Scrum-Workflow

eingegangen. Wie bereits im OpenUP-Prozess wird auch hier auf die Rollen, der am Prozess beteiligten Personen, nur bedingt eingegangen.

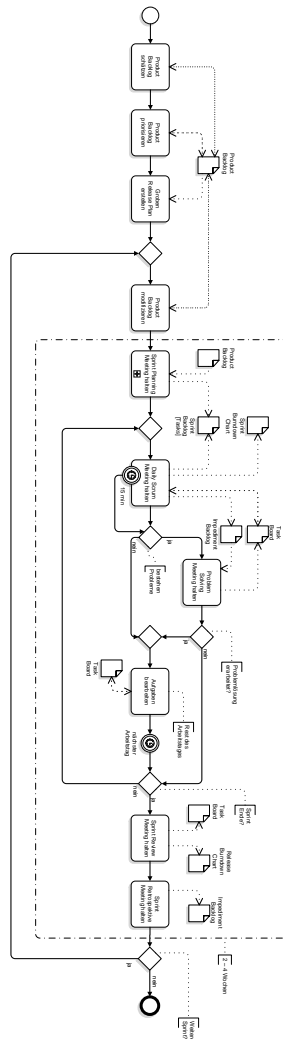


Abbildung 5.15: Scrum: Workflow, Quelle: [Men14] mit eigenen Anpassungen

Die Nachfolgende Beschreibung bezieht sich auf den Prozess, welcher in Abbildung 5.15 komplett dargestellt ist. In den Abbildungen 5.16 und 5.17 sind einzelne Bereiche des Prozesses hervorgehoben. Bevor mit der eigentlichen Arbeit begonnen werden kann, muss das Team sich mit dem Product Backlog vertraut machen und auf dessen Basis den Zeitaufwand abschätzen, um das Projekt abzuschließen. Diese Abschätzung bildet die Grundlage zum Priorisieren des Backlogs. Auf Basis dessen wird wiederum, in Verbindung mit den zeitlichen Abschätzungen, der grobe Release Plan erstellt. Mittels dieses Plans werden grobe Feinheiten des Projekts beschrieben, wie zu Beispiel die Anzahl der Teams und deren Mitglieder sowie die Menge der Sprints. Die Dauer eines solchen Sprints ist immer identisch und liegt meist bei 4 Wochen.

Die nachfolgenden Schritte sind in Abbildung 5.17 hervorgehoben. Jeder Sprint beginnt mit einem *Sprint Planning Meeting*. Dieses ist ein eigener Unterprozess, welcher aus zwei Meetings besteht, auf Grund der Einfachheit dieses Subprozesses wurde hier auf eine explizite Modellierung verzichtet und bei der folgenden textuellen Beschreibung belassen. Im ersten der beiden Meetings werden die am höchsten priorisierten Einträge des Product Backlogs erläutert und daraus ein Ziel für den bevorstehenden Sprint definiert. Dieses zuvor festgelegte Ziel wird beim zweiten Meeting vom entsprechenden Team in konkrete Aufgaben ausgearbeitet, welche anschließend in den Product Backlog aufgenommen werden.

Jeder Arbeitstag während eines laufenden Sprint beginnt mit dem so genannten Daily Scrum Meeting. Dieses Meeting findet stets zur selben Zeit und am gleichen Ort statt. Es darf zwar jeder daran teilnehmen, allerdings haben nur Scrum-Team Mitglieder Sprachrecht. Bei diesen Meetings muss jedes Mitglied sich zu drei zentralen Fragestellungen äußern. Zum Einen muss er offenlegen, was er seit dem letzten Meeting getan hat und was er bis zum nächsten Meeting schaffen wird. Und zum Anderen, ob es Gründe (so genannte Imperdiments) gibt, die ihn in seiner Arbeit behindern. Diese werden anschließend schriftlich festgehalten. Ein solches Meeting hat eine maximale Dauer von 15 Minuten. Sollten sich im täglichen Meeting Probleme gezeigt haben, so wird im Anschluss daran ein Problem Solving Meeting abgehalten. Dieses dient wie bereits erwähnt dazu die während des täglichen Meetings aufgezeigten Probleme zu diskutieren und zu lösen. Sofern diese gelöst werden können, wird für den Rest des Arbeitstages an den definierten Aufgaben gearbeitet. Der nächste Arbeitstag beginnt abermals mit einem täglichen Meeting. Falls die Probleme nicht gelöst werden können, muss der Sprint abgebrochen werden. Am Ende eines Sprints oder falls der Sprint auf Grund von Problemen beendet werden musste, wird ein sogenanntes *Sprint Review Meeting* abgehalten. Bei diesem werden die Ergebnisse des Sprints und die erarbeiteten Funktionalitäten präsentiert. Zudem wird hier die Entscheidung gefällt, ob der Sprint erfolgreich gewesen ist. Im Anschluss daran trifft sich das Team, um den Sprint retrospektiv zu betrachten. Es wird geprüft, was gut gelaufen ist oder welche Probleme auftraten. Letztere werden im Imperdiment-Log festgehalten. Zudem wird festgestellt, was man bei diesem Sprint gelernt hat und was man in die Zukunft mitnehmen möchte oder auch was man vermeiden sollte. Dieses Meeting markiert das Ende des Sprints und es kann entschieden werden, ob ein weiterer Sprint nötig ist. Trifft dies zu, wird wiederum der Project Backlog modifiziert und der neue Sprint kann beginnen. Falls kein weiterer Sprint nötig ist, wird der Prozess erfolgreich beendet.

Wie auch bei den Prozessen zuvor, wird auch der Scrum-Prozess auf kontextuelle Einflüsse analysiert. Eine der wichtigsten Eigenschaften von Scrum ist, dass das Team während eines laufenden Sprints die Anforderungen an das System nicht geändert werden dürfen, weswegen der Prozess in Haupt- und Subprozess aufgeteilt wird. Ebenso erfolgt die Analyse des Prozesses. Die Aufteilung ist in Abbildung 5.16 modelliert. Die Analyse beginnt mit dem Hauptprozess, wobei der Subprozess *Scrum-Sprint* hier ausgelassen und daraufhin separat analysiert wird.

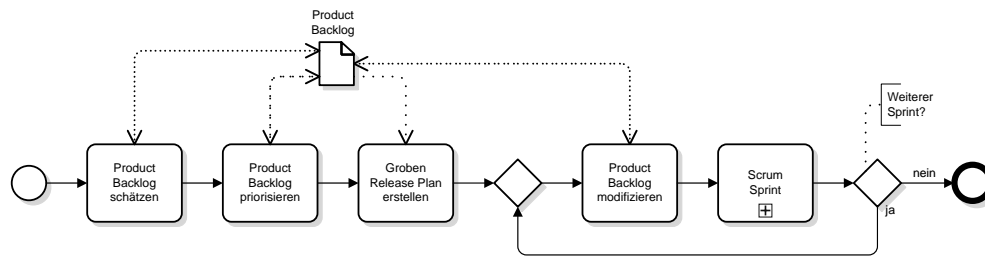


Abbildung 5.16: Scrum: Workflow (verkürzt), Quelle: [Men14] mit eigenen Anpassungen

Betrachtet man die vier Aktivitäten, welche dem Sprint vorangehen, unter dem oben beschriebenen Gesichtspunkt (keine Änderungen während des Sprints), kommt man zu dem Ergebnis, dass alle anfallenden Änderungen hier aufschlagen müssen. Bereits im ersten Schritt des Prozesses kann es passieren, dass sich die Anforderungen ändern. Dies hätte unter Umständen zur Folge, dass die getätigten Schätzungen nicht mehr korrekt sind. Als weiterer Einfluss kann es bei einem frisch gestarteten Projekt vorkommen, dass sich bei einem bereits fertiggestellten Produkt dringende Bugfixes ergeben haben, wodurch ein Teil des geplanten Projektteams fürs erste entfällt. Auch dies führt wiederum dazu, dass die Schätzungen angepasst werden müssen. Ähnliche Einflüsse ergeben sich auch in der zweiten Aktivität und führen zur Anpassung der Prioritäten. Ebenso einschlägig sind kontextuelle Einflüsse auf die Erstellung des Release-Plans. Hier kann das Wegfallen eines Kollegen, zum Beispiel auf Grund von Krankheit oder einer aufgetretenen Ad-hoc-Aktivität eines anderen Projekts, zu Verzögerungen führen. Solche Kontextänderungen vor dem Start der eigentlichen Entwicklungsarbeit sind zwar ärgerlich jedoch kompensierbar. Schlimmer wird es, wenn das Projekt bereits angelaufen ist und ein oder auch mehrere Sprints erfolgreich abgeschlossen wurden. Die Aktivität *Product Backlog modifizieren* muss alle kontextuellen Einflüsse auf den Prozess abfangen. Hier muss der Projektrückstand stets den sich verändernden Gegebenheiten angepasst werden, weswegen diese Aktivität, kontextuell gesehen, der Dreh- und Angelpunkt des gesamten Prozesses ist.

Wie bereits in der Prozessbeschreibung erwähnt, dürfen die Anforderungen an das System während eines laufenden Sprints nicht geändert werden. Ebenso sollte nach Möglichkeit jede andere Störung des Sprint-Teams von außen vermieden werden. Dass dies in konkreten Fällen nicht immer möglich sein wird, soll an dieser Stelle vernachlässigt werden. Die nachfolgende Analyse bezieht sich auf die Abbildung 5.17, welche den Scrum Sprint nochmals hervorhebt.

Jeder neue Sprint muss sorgfältig geplant werden, dies passiert beim *Sprint Planning Meeting*. Diese Aktivität ist zudem die letzte, bei welcher kontextueller Einfluss von „außen“ auf den Prozess Auswirkung nehmen kann. Diese Einflüsse sollten allerdings nach Möglichkeit vermieden werden. Wie bereits eingehend erwähnt, basiert Scrum unter anderem auf starker Kommunikation zwischen den Team-Mitgliedern. Eine Aktivität, welche diesen Aspekt fördert, ist das *Daily*

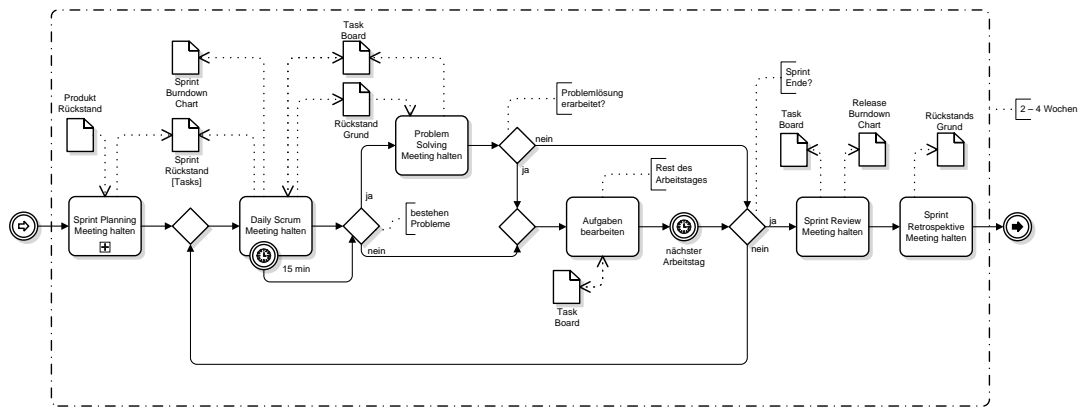


Abbildung 5.17: Scrum: Workflow (Sprint), Quelle: [Men14] mit eigenen Anpassungen

Scrum Meeting. Dieses Meeting bietet zudem die Möglichkeit veränderten Kontext zu registrieren und zu entscheiden, wie darauf reagiert werden soll. Entsteht dadurch ein Problem wird dieses schriftlich festgehalten und im anschließenden *Problem Solving Meeting* ausdiskutiert. An dieser Stelle des Prozesses werden eventuell auftretende Kontextänderungen weitestgehend ignoriert, da sie nicht zur Lösung des Problems beitragen. Die nächste kontextuell beeinflussbare Aktivität ist die eigentliche Arbeit der Entwickler. Wenn auch die Kontextänderung hier meist in einem sehr kleinen Rahmen auftritt. Beispiele hierfür wären eine unerwartete Erkrankung oder Verletzung des Entwicklers oder technische Störungen seines Arbeitsgeräts.

Alles in allem ist Scrum ein Prozess, welcher durch seine Basisidee relativ kontextunabhängig ist.

5.3 Diskussion möglicher Alternativen

Die BPMN Syntax erlaubt die Modellierung von Kontext mittels unterschiedlicher Möglichkeiten, welche untersucht und bewertet werden müssen. Jede dieser Alternativen hat in der BPMN einen Verwendungszweck, welcher hier exemplarisch genutzt wird um damit kontextuelle Einflüsse abzubilden. Diese wären:

- Datenobjekte und Datenfluss
- exklusives Gateways
- inklusives Gateway
- komplexe Gateways
- Ereignis-basierte Gateways
- angeheftete Ereignisse (zum Beispiel Nachricht, Bedingung oder Signal)
- benutzerdefinierte Artefakte

- eine Kombination aus bereits genannten

Wie praktikabel diese Möglichkeiten sind, wird in diesem Abschnitt anhand mehrerer Beispiele untersucht. Jede Möglichkeit wird hierzu modelliert und bewertet, indem die Stärken und Schwachpunkte hervorgehoben und gegeneinander abgewogen werden.

5.3.1 Datenobjekte und Datenfluss

Als erstes Beispiel eignet sich der bereits in der Einleitung beschriebene Prozess der Diagnoseerstellung. Dieser ist in Abbildung 5.18 modelliert zu sehen.

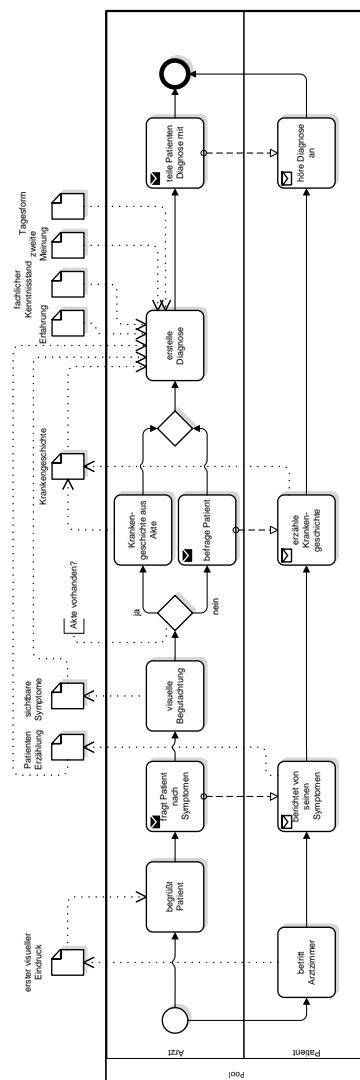


Abbildung 5.18: Kontexteinfluss bei Diagnoseerstellung (Datenobjekte)

Man kann bereits an diesem Minimalbeispiel erkennen, dass bei stark kontextsensitiven Prozessen die Anzahl der Datenelemente und Datenkanten sehr schnell zu unübersichtlichen Modellen

führt. Was dazu führen kann, dass diese in der Modellierung vernachlässigt werden, wodurch die Kontextsensitivität im Modell verloren geht. Zudem ist es schwer „normale“ Datenelemente von den den Kontext beschreibenden zu unterscheiden, ohne letztere explizit zu kennzeichnen. Dies würde jedoch, besonders bei Prozessen mit hohem Datenaufkommen, die Übersichtlichkeit weiter herabsetzen und das Design unnötig verkomplizieren.

5.3.2 Exklusive Gateways

Die Verwendung von exklusiven Gateways ist eine intuitive Alternative zur Modellierung von Kontext. Dieser Gatewaytyp eignet sich hervorragend, um eine aus mehreren Alternativen zu wählen. Eine der Schwächen dieser Alternative ist, dass stets genau einer der möglichen Pfade gewählt wird. Sollte die Prozessdefinition es erfordern, dass mehrere Schritte parallel ausgeführt werden, müssen diese nun durch ein paralleles Gateway zusammengefasst werden. Wie schnell dies zu äußerst schlecht durchschaubaren und damit wenig einsetzbaren Prozessmodellen führt, ist in Abbildung 5.19 exemplarisch dargestellt. In diesem Beispiel sind lediglich vier verschiedene Aktivitäten mit allen Kombinationen aus zwei parallelen oder einer einzelnen Aktivität dargestellt. Mögliche Kombinationen in diesem Beispiel wären unter anderem: *A*, *B*, *C*, *D*, *AB*, *AC*, *BD*. Würde man nun zudem alle möglichen Kombinationen mit drei Aktivitäten einfügen oder gar die Anzahl dieser erhöhen, wäre das Modell für einen Menschen nicht mehr lesbar, da die einzelnen Pfade sich zu sehr überlappen und deswegen nicht mehr nachverfolgbar sind. Für die Möglichkeit der automatisierten Prozessausführung wäre diese Alternative allerdings nutzbar. Dies lässt sich auch mathematisch zeigen. Die Zahl der Möglichkeiten n Elemente aus der Menge k auszuwählen entspricht dem Binomialkoeffizienten, diese Rechnung müsste man für alle n durchführen und die Ergebnisse addieren. Dies würde im gegebenen Beispiel bedeuten dass 13 verschiedene Kombinationen abgebildet werden müssten. Erhöht man die Anzahl der Aktivitäten um eins, so sind es bereits 31 Kombinationen. Zwar lässt sich diese Zahl durch geschickte Verwendung von Gateways verringern, die Übersichtlichkeit wird durch diese Möglichkeit jedoch nicht erhöht. Dieses einfache Beispiel schließt die Verwendung von exklusiven Gateways zur Modellierung von Kontextabhängigkeit als allgemeine Alternative aus. Für sehr kleine Anwendungen ist es jedoch, auf Grund der intuitiven Verwendbarkeit, durchaus geeignet.

5.3.3 Inklusive Gateways

Diese Gateway Typen gleichen die Schwäche des zuvor beschriebenen exklusiven Gateways aus. Sie ermöglichen es k aus n Pfaden auszuwählen, wodurch die gewünschte Semantik jetzt mit nur einem anstatt 5 (siehe Abbildung 5.19) Gateways modellieren können. Der Prozessgraph wird dadurch zwar ein wenig kompakter, jedoch nicht zwangsläufig übersichtlicher. Abbildung 5.20 verdeutlicht diesen Umstand. Zur besseren Übersicht ist der Prozessverlauf in dieser Ab-

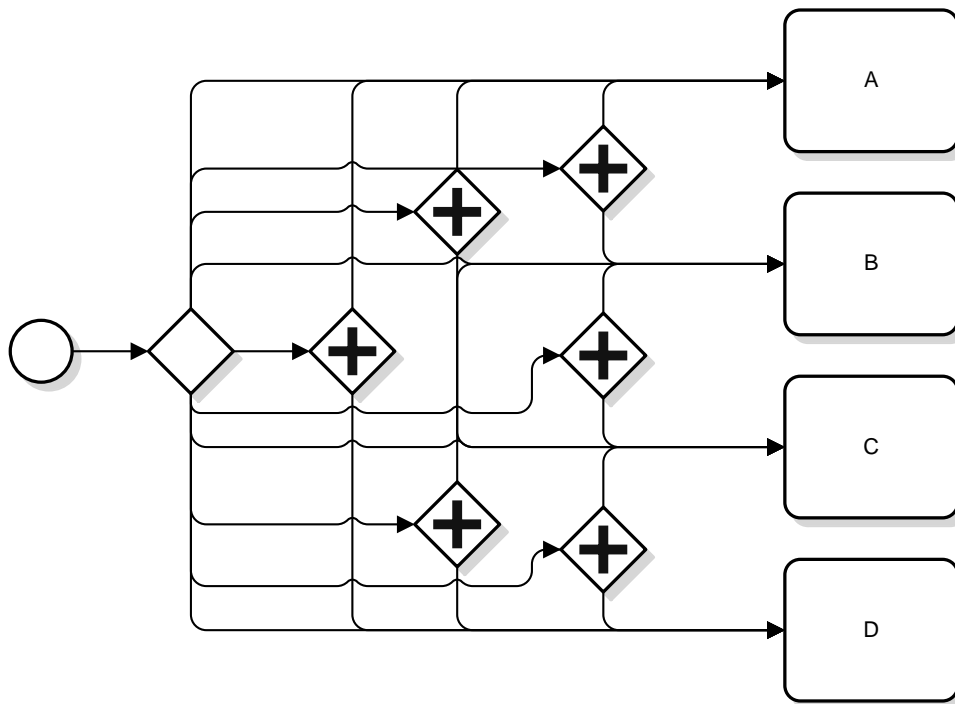


Abbildung 5.19: Beispiel für die Komplexität von exklusiven Gateways

bildung von oben nach unten und nicht von links nach rechts wie bei den anderen Prozessen in dieser Arbeit. Die Abbildung zeigt die Kombination aus neun verschiedenen Aktivitäten, verbunden durch ein inklusives Gateway. Welche dieser Kombinationen ausgeführt werden, wird anhand von Daten entschieden welche an früheren Punkten der Prozessausführung gesetzt wurden entschieden. Je mehr Aktivitäten dazukommen, desto größer und unübersichtlicher wird das Modell. Dies kann vor allem bei Prozessen mit vielen kontextuell beeinflussten Aktivitäten mit einer Vielzahl von Alternativen zu komplexen Modellen führen. Daraus lässt sich folgern, dass diese Alternative sich kleine Modelle mit geringer Varietät eignet. Bei großen und komplexen Anwendungen ist sie allerdings nicht zu empfehlen.

5.3.4 Komplexe Gateways

Dieses Gateway ist hervorragend geeignet um komplexe Join Prädikate darzustellen. Ein Beispiel für eine solche Anwendung ist in Abbildung 5.21 zu finden. In diesem Beispiel kann die Aktivität D erst gestartet werden sobald 2 Aktivitäten aus der Menge der Aktivitäten A, B und C beendet wurden. Angewandt auf die Kontext-Thematik ermöglicht die Verwendung dieses Gateways eine Verringerung der Komplexität beim Zusammenführen mehrerer Pfade.

Wie jedes andere auch, lässt sich das komplexe Gateway als Split verwenden. In einem solchen Fall wäre es möglich ein oder auch mehrere exklusive Gateways ersetzen. Dies hätte eine Verringerung der Pfade zur Folge, was die Komplexität herabstuft. Allerdings erfordert dies ei-

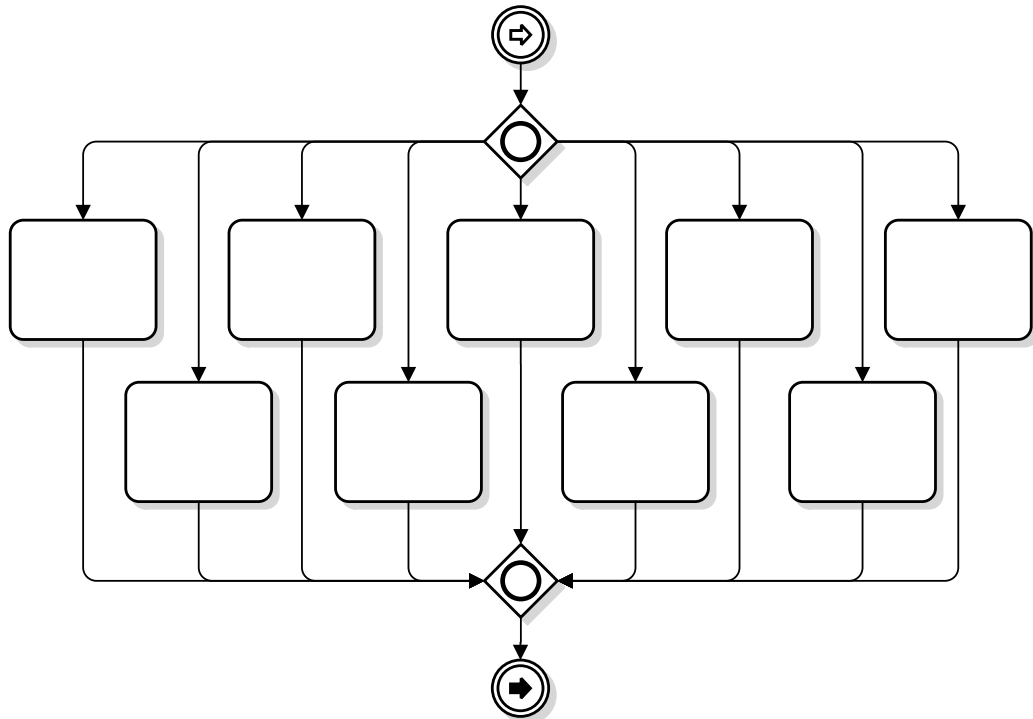


Abbildung 5.20: Beispiel für die Komplexität von inklusiven Gateways

ne ausführliche Dokumentation mittels Anotationen. Würden diese weggelassen werden, ist die Ausführungssemantik des Modells nicht mehr nachvollziehbar und das Modell nur noch eingeschränkt nutzbar.

5.3.5 Ereignisbasierte Gateways

In Kapitel 2.1.2 wurde die Funktionalität des ereignisbasierten Gateways und dem Unterschied zu den datenbasierten Varianten erläutert. Beide Typen, parallel und exklusiv, haben jedoch eindeutige Schwächen bei der Kontextmodellierung.

Den Anfang macht das parallele Gateway. Die ereignisbasierte Variante existiert einzig als Prozessstartsymbol, zu sehen in Abbildung 5.22. Hier kann es dazu verwendet werden, mehrere Ereignisse zu korrelieren und dadurch den Prozess starten zu lassen. Dies ist an bestimmten Stellen eine äußerst nützliche Eigenschaft, reicht jedoch bei weitem nicht aus, um Kontext im Prozessfluss zu modellieren.

Auch der zweite Gateway-Typ, welcher hier diskutiert wird, verfügt über die Möglichkeit es zur Prozess-Instanziierung zu verwenden. Diese wird exemplarisch in Abbildung 5.23 dargestellt.

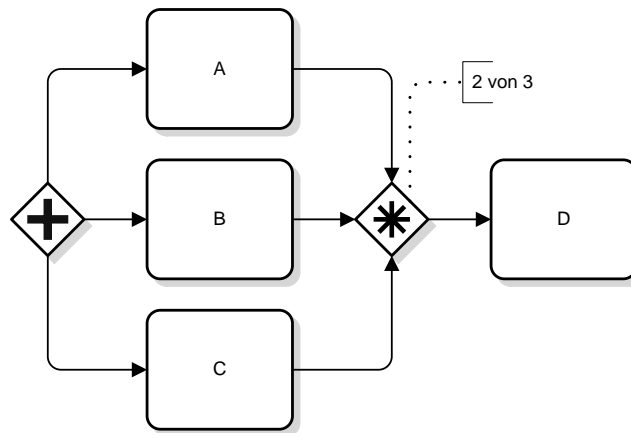


Abbildung 5.21: Beispiel für die Verwendung von komplexen Gateways

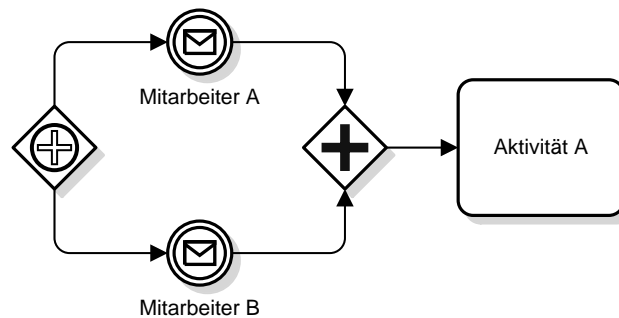


Abbildung 5.22: Beispiel für die Verwendung von ereignisbasierten parallelen Gateways

Hierbei muss beachtet werden, dass die Verwendung des Gateways in diesem Fall nicht zwingend notwendig ist. Es ist ebenso möglich dieselbe Semantik mittels zweier Nachrichten-Startevents und einem exklusiven Join-Gateway zu realisieren. Das zusätzliche Gateway trägt in diesem Fall nicht zur Übersichtlichkeit bei, sondern die Lesbarkeit des Prozesses und erhöht unnötig die Komplexität. Daraus lässt sich folgern, dass auch dies keine wirkliche Alternative zur Kontextmodellierung darstellt. Zumal diese Alternative lediglich zur Modellierung des Prozessstarts verwendet werden kann.

Allerdings lässt sich dieses Gateway, im Gegensatz zum datenbasierten parallelen, an jeder beliebigen Stelle im Prozess verwenden. Die Auswahl der akzeptierten Ereignisse ist jedoch eingeschränkt. Eine Übersicht ist in Abbildung 5.24 aufgezeigt. Jede dieser Kombinationen hat ihre Stärken und Schwächen und muss daher gesondert betrachtet werden. Die Reihenfolge der Analysen entspricht der Abbildung.

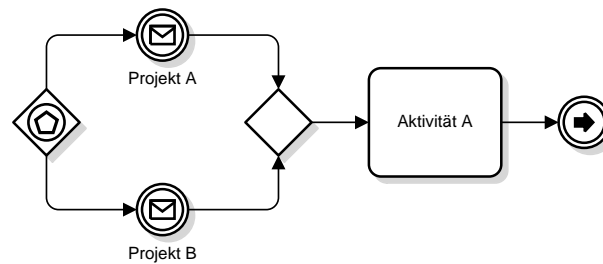


Abbildung 5.23: Beispiel für die Verwendung von ereignisbasierten exklusiven Gateways zum Prozessstart

Nachricht

Die Steuerung des Prozessflusses mittels Nachrichten ist eine elegante und komfortable Möglichkeit auf Ereignisse zu reagieren. Zumindest das Nachrichtenereignis in der BPMN sehr generisch definiert ist und stellvertretend für Vieles genommen werden kann. Daraus lässt sich folgern, dass diese Kombination gut geeignet ist, um das gewünschte Ziel zu erreichen, die Modellierung von Kontext. Das Problem, das hier vorliegt, ist das selbe wie bei anderen Alternativen - die Komplexität. Solange sich die Menge der möglichen Nachrichten in Grenzen hält, ist dies eine anwendbare Alternative. Mit steigender Anzahl der möglichen Nachrichten wird diese Schwachstelle schwerwiegender. Ein weiterer Schwachpunkt ist die generische Verwendung des Nachrichtentyps. Da eine Nachricht per se alles sein kann, steigert es die Komplexität in den Modellen um ein Vielfaches. Zudem muss beachtet werden, dass eine Nachricht in BPMN immer einen festgelegten Empfänger besitzen muss. Dies ist im Kontextfall nicht immer gegeben. Daraus lässt sich folgern, dass dies nicht die gesuchte Alternative darstellt.

Zeit

Das Zeitereignis in BPMN ist äußerst flexibel und lässt sich sowohl für Intervalle als auch für festgelegte Zeitpunkte verwenden. In Kombination mit dem ereignisbasierten Gateway lassen sich zeitgesteuerte Prozessflüsse definieren. Zeit ist ein wichtiger Faktor bei Kontextbetrachtung, jedoch nicht der einzige. Zudem muss für den Fall, dass mehrere zeitgesteuerte kontextuelle Einflüsse existieren, für jeden Einfluss ein Prozesspfad modelliert werden. Da, wie bereits erwähnt, Zeit nicht der einzige Kontextfaktor ist, reicht diese Kombination nicht aus, um alle kontextuellen Einflüsse zu modellieren. Somit ist diese Kombination für die Kontextmodellierung zwar sehr interessant, jedoch keine Alternative.

Signal

Das Signalereignis in BPMN ist beinahe identisch zu dem Nachrichtenereignis. Der Unterschied liegt darin, dass das Signalereignis keinen expliziten Adressat hat. Somit fällt diese Schwachstelle weg. Es muss allerdings noch immer für jedes mögliche Signal ein Pfad modelliert werden, was bei einer hohen Anzahl möglicher Signale sehr unübersichtlich wird.

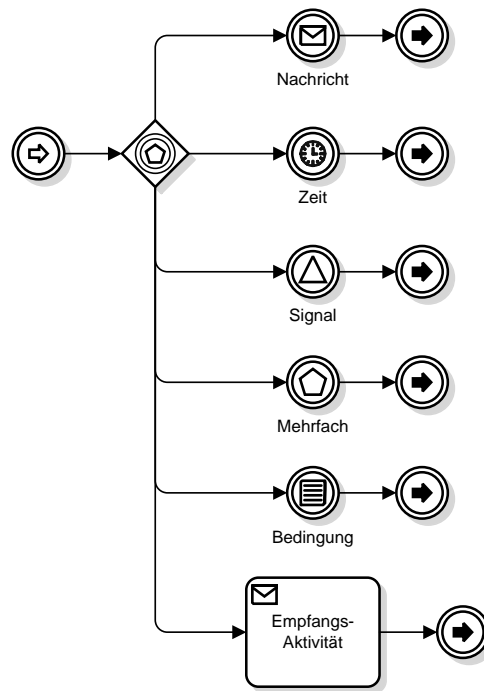


Abbildung 5.24: Beispiel für die Verwendung von ereignisbasierten exklusiven Gateways

Zudem muss bei der Verwendung dieser Kombination darauf geachtet werden dass man sich strikt an die Vorgabe hält, dass ein Signal keinen festgelegten Empfänger hat. Fasst man dies zusammen ergibt sich der folgende Sachverhalt. Das Signalereignis ist bis auf einen Unterschied identisch zum Nachrichtenereignis. Es besitzt keinen festgelegten Empfänger, was eine der Schwächen des Nachrichtenereignisses war. Die Schwäche dieses Ereignisses ist, dass die Anzahl der Pfade der unterschiedlichen Signale entspricht. Dies hat zur Folge, dass die Modelle, bei entsprechender Signalvielfzahl, sehr unübersichtlich und komplex werden. Diese Schwäche macht die Kombination für die allgemeine Kontextmodellierung nicht anwendbar.

Mehrfach

Diese in der Praxis eher unübliche Kombination verfügt über eine interessante Eigenschaft. Es fasst mehrere unterschiedliche Ereignisse zusammen. Was jedoch im Umkehrschluss bedeutet, dass es sehr ausführlich dokumentiert werden muss, da es nicht intuitiv verständlich ist. Trotzdem hat diese Kombination einen Vorteil, die damit modellierten Prozessmodelle sind sehr kompakt und übersichtlich. Jedoch auch unlesbar und so gut wie nicht verständlich, ohne die dazugehörige Dokumentation zu studieren. Dies bedeutet, dass die Komplexität des Modells lediglich ausgelagert wird. Sollte diese Kombination mehrfach im Prozess eingesetzt werden, ist es unter Umständen einfacher, die Dokumentation zu studieren und auch den Rest des Prozesses in diese aufzunehmen. Sollte dieser Fall eintreten, ist das Erstellen eines BPMN Modells nicht notwendig und erzeugt des-

wegen unnötige Kosten. Da jedoch genau dies das angestrebte Ziel ist, unter anderem um die Notwendigkeit einer textuellen Dokumentation zu reduzieren beziehungsweise diese möglichst kompakt zu halten, ist die Kombination aus ereignisbasiertem Gateway und Mehrfachereignis für die Darstellung von Kontext nicht anwendbar.

Bedingung

Bei dieser Kombination ist zu beachten, dass die abgefragte Bedingung unabhängig vom Prozess erfüllt werden muss. Zudem gilt hier, wie bei vielen anderen Konstrukten, dass beinahe alles eine Bedingung sein kann, solange diese außerhalb des Prozesses erfüllt wird. Dies eröffnet ein weites Feld von Möglichkeiten zur Prozesssteuerung. Zu beachten ist jedoch, dass das Erfüllen der Bedingung den Pfad aktiviert. Sofern dies Element zur Ablaufsteuerung verwendet wird, was bei Kontextbetrachtung durchaus der Fall ist, so muss der Pfad, bei welchem die Bedingung nicht erfüllt wurde, explizit modelliert werden. Dies bedeutet im konkreten Fall, dass bei n Bedingungen $2n$ Prozesspfade modelliert werden müssen. Sofern ein Standardpfad benötigt wird, muss dieser ebenfalls mit modelliert werden. Was abermals zu dem Problem der fehlenden Übersichtlichkeit durch eine hohe Menge an Pfaden führt. Für sich allein genommen ist diese Kombination ebenfalls keine praxistaugliche Alternative.

Empfangsaktivität

Die Kombination von ereignisgesteuertem Gateway und Empfangsaktivität entspricht semantisch einem Nachrichtenereignis gefolgt von einer Aktivität und stellt somit lediglich eine verkürzte Notation dar. Auf Basis dieser Tatsache wird diese hier nicht nochmals diskutiert. Die Erwähnung erfolgt nur aus Gründen der Vollständigkeit. Für eine Analyse der Anwendbarkeit zur Kontextmodellierung sei an dieser Stelle auf die Analyse des entsprechenden Ereignisses weiter vorne verwiesen.

Nach der Analyse der einzelnen Variationen muss nun die Möglichkeit der Kombination unterschiedlicher Ereignisse betrachtet werden. Auf die Aufzählung aller möglichen Kombinationen sei an dieser Stelle verzichtet, da es praktisch keine Einschränkung dafür gibt. Es wird einzig auf die Stärken und Schwächen der Kombination eingegangen und die Verwendbarkeit zur Kontextmodellierung betrachtet. Durch die Kombination unterschiedlicher Ereignistypen lassen sich äußerst feingranulare Prozessflüsse modellieren. Grundvoraussetzung hierfür ist die genaue Kenntnis der möglichen Ereignisse. Zudem kumulieren die Vorteile der einzelnen Ereignistypen bei deren Kombination. Dies gilt jedoch ebenso für die Nachteile und Schwächen der einzelnen Möglichkeiten. Kombiniert man zum Beispiel zwei Alternativen, welche zu unübersichtlichen Prozessen führen, kann der daraus resultierende Prozess noch unübersichtlicher werden. Sodass deren Summe, je nach Anwendung, überwiegen kann. Im Fall der Kontextmodellierung ist dies der Fall. Daraus lässt sich folgern, dass dieses Element ebenfalls nicht ausreicht, um Kontext in Prozessen zu modellieren.



Abbildung 5.25: Betrachtete Ereignistypen

5.3.6 angeheftete Ereignisse

Eine Alternative, welche auf den ersten Blick immense Möglichkeiten eröffnet, sind die angehefteten Ereignisse. Sie bieten beinahe die gesamte Bandbreite der in der BPMN definierten Ereignistypen und lassen sich zudem in unterbrechende und nicht unterbrechende angeheftete Ereignisse unterteilen. Im Folgenden werden die einzelnen Typen als unterbrechendes und als nicht unterbrechendes angeheftetes Ereignis einzeln betrachtet und auf ihre Verwendbarkeit untersucht. Eine Kombination dieser mit anderen BPMN Konstrukten wird an dieser Stelle ausgelassen, da sie im Kapitel 5.3.8 eingehend betrachtet werden. Eine Übersicht über die betrachteten Ereignistypen ist in Abbildung 5.25 gegeben.

Den Anfang macht das Nachrichten-Ereignis. In Abbildung 5.26 wurde ein solches Ereignis verwendet, um sich verändernden Kontext und die daraus resultierenden Aktivitäten zu modellieren. Das Beispiel stellt einen Auszug aus einem möglichen größeren Softwareentwicklungsprozess dar. Der Akteur bearbeitet die Aufgabe *Meilenstein implementieren*, als eine Nachricht von einem Kollegen eintrifft, welche ihn darüber in Kenntnis setzt, dass die Konvention zur Benennung der Variablen geändert wurde. Diese Nachricht muss der Akteur zwar zur Kenntnis nehmen und den daraus resultierenden Schritt, die Umbenennung von bereits verwendeten Variablen, zwar ausführen, muss dafür jedoch nicht seine aktuelle Arbeit unterbrechen.

Ganz anders sieht es in Abbildung 5.27 aus. In diesem Beispiel informiert die ankommende Nachricht über eine wichtige Änderung in den Anforderungen, was ein sofortiges Meeting erfordert. Hierfür muss die Bearbeitung der aktuellen Aufgabe abgebrochen werden.

Die erste Schwäche dieser Möglichkeit lässt sich direkt aus den zuvor verwendeten Beispielen ableiten. Die Entscheidung, ob das angeheftete Ereignis den Prozessschritt unterbrechen soll oder nicht, hängt häufig von der eingehenden Nachricht ab. Entfernt man sich von der reinen Modellierungsgedanken, kann man allerdings argumentieren, dass das Lesen einer Nachricht, unabhängig von ihrem Inhalt, immer zu einer Unterbrechung der Aktivität führt. Dies ist soweit zwar korrekt, trifft jedoch nicht den Kern dessen, was der Standard als unterbrechend definiert. In dem meisten Fällen wird die Bearbeitung der Aktivität unterbrochen, um die eingegangene

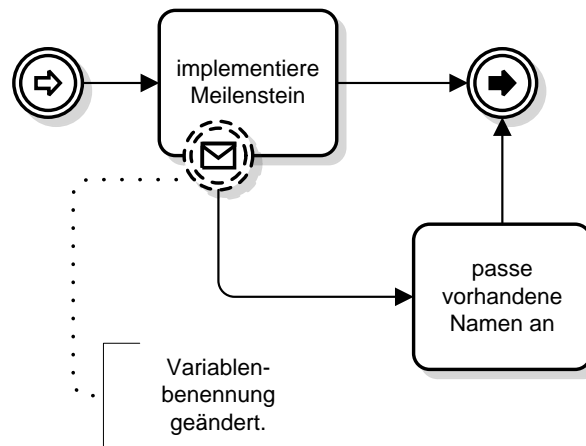


Abbildung 5.26: Beispiel für nicht unterbrechendes angeheftetes Nachrichtenereignis

Nachricht zu lesen. Der Unterschied liegt darin, was danach geschieht. Kehren wir zur Bearbeitung der Aktivität zurück, so gilt diese als nicht unterbrochen. Die zuvor genannte Entscheidung lässt sich jedoch nicht mit den von der BPMN definierten Mitteln modellieren, was diese Alternative nicht brauchbar macht um kontextuelle Einflüsse darzustellen. Die zweite Schwäche teilt sich das Nachrichten Ereignis mit allen anderen angehefteten Ereignissen und wird deswegen am Ende dieses Kapitels beschrieben.

Das nächste angehängte Ereignis, welches hier betrachtet wird, ist die Bedingung. Hierbei wird eine Bedingung, welche außerhalb des Einflussbereiches des Prozesses erfüllt werden muss, ausgewertet. Ein, an sich, vielfältiges Werkzeug zur Prozesssteuerung. Die Schwäche liegt hier in der Definition. Die Bedingung muss zwingend außerhalb des Prozesses erfüllt werden. Dies ist bei Kontextänderungen nicht immer gegeben! Vor allem unter dem Gesichtspunkt, dass in der vorliegenden Arbeit der Kontext nicht als Rahmenbedingung des Prozesses gesehen wird. Aus den genannten Gründen ist die Verwendung des Bedingungsereignisses als angehängtes Ereignis nicht durchgehend geeignet um Kontext darzustellen.

Fortgesetzt wird die Reihe durch einen Ereignistyp, welcher dem Nachrichtenereignis stark ähnelt. Dem Signalereignis. Wie bereits in Kapitel 5.3.5 beschrieben, ist der Hauptunterschied, zwischen den beiden Ereignistypen der Empfänger. Während eine Nachricht zielgerichtet ist und einen festgelegten Empfänger hat, entspricht ein Signal eher einer Zeitungsanzeige deren Empfänger derjenige ist, wer auf die Anzeige als erster reagiert. Es wird ausgelöst und jemand reagiert darauf. Zu Beginn des Abschnittes wurde eine Schwäche des Nachrichtenereignisses beschrieben, die Auswertung, ob es unterbrechend ist oder nicht zum Zeitpunkt der Modellierung. Diese Schwäche existiert beim Signalereignis ebenfalls.

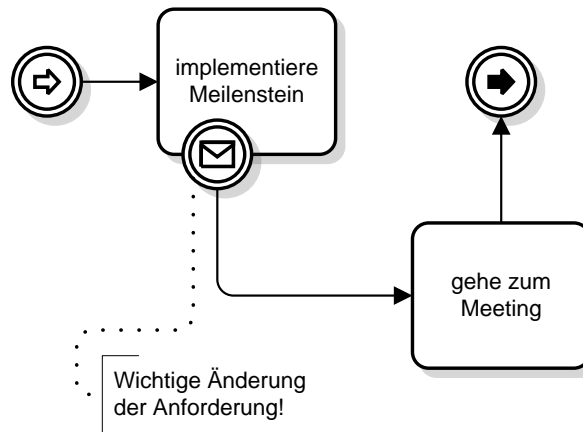


Abbildung 5.27: Beispiel für unterbrechendes angeheftetes Nachrichtenereignis

Ein Ereignis, welches mehr der Vollständigkeit halber hier aufgenommen wurde, ist das Eskalationsereignis. Das Ereignis dient der Kommunikation zwischen Subprozess und Oberprozess. Zur Modellierung von Kontext daher gänzlich ungeeignet.

Auch das nächste Ereignis wurde bereits in einem anderen Zusammenhang betrachtet. Es handelt sich hierbei um das Mehrfachereignis. Und auch hier ergibt sich die selbe Schwachstelle, es ist nicht intuitiv und muss ausführlichst dokumentiert werden. Daraus ergibt sich die Frage, ob es nicht einfacher ist, den gesamten Prozess textuell zu beschreiben. Und auch an dieser Stelle ist dies nicht das angestrebte Ziel. Weswegen von der Verwendung des Mehrfachereignisses zur Kontextdarstellung an dieser Stelle abgeraten sei.

Zu Beginn dieses Kapitels wurde angesprochen, dass alle angehängten Ereignisse einen gemeinsamen Schwachpunkt besitzen, welcher nun genauer betrachtet werden soll. Der erste Punkt, welcher in diesem Zusammenhang angesprochen werden muss, ist die Bedeutung der Symbole. Sie sind meist sehr generisch und haben daher sehr viele Deutungsmöglichkeiten. Daraus folgt, dass die textuelle Dokumentation der Prozesse äußerst umfangreich sein muss, was, wie bereits erwähnt, nicht zielführend ist. Zudem die einzelnen Symbole für sich allein oft nicht ausreichend sind, um komplexe Modelle darzustellen. Erst die Kombination mit anderen Symbolen ermöglichen es alle benötigten Situationen zu darzustellen. Dieser Umstand kann jedoch zu sehr komplexen und unübersichtlichen Prozessmodellen führen, welche wiederum eine umfangreiche textuelle Dokumentation erforderlich machen.

5.3.7 benutzerdefinierte Artefakte

Die Möglichkeit benutzerdefinierte Artefakte in die Prozessmodelle einzubauen bietet eine Alternative, deren genauere Betrachtung, für die gegebene Fragestellung, lohnenswert ist. Es ist dadurch möglich, kontextuell interessante Aktivitäten mit einem aussagekräftigen Artefakt zu verbinden und dieses damit eindeutig zu kennzeichnen. Es wäre, mittels dieser Artefakte, daher möglich, kontextuellen Einfluss klar und sauber von Datenelementen zu trennen. In Abbildung 5.28 wurde ein mögliches Artefakt als Beispiel modelliert.

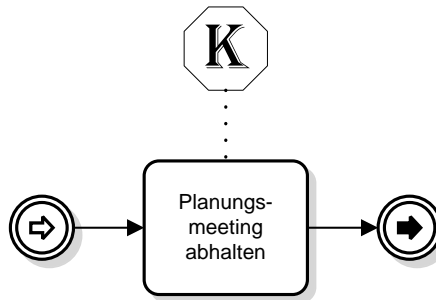


Abbildung 5.28: Beispiel für eigenes Artefakt

Statt neue Symbole einzuführen, könnte das eine Alternative sein. Wie allerdings der Name schon sagt, handelt es sich hierbei lediglich um Artefakte. Damit ist es zwar möglich, auf Kontext bei einer Aktivität hinzuweisen, jedoch nicht diesen auch zu modellieren. Zudem können Modelle, welche mit eigenen Artefakten versehen wurden, nicht von anderen gelesen werden, da dieser Person die Artefakte und deren Bedeutung nicht bekannt sind.

5.3.8 Kombinationen aus verschiedenen Alternativen

Wie bereits erwähnt, ist die umfangreichste Alternative die Kombination aus mehreren zuvor diskutierten. Die Stärken der einzelnen Symbole summieren sich, allerdings auch die Schwächen. Ein Beispiel für solch eine Kombination sind die ereignisbasierten Gateways. Es ist jedoch auch jede andere, syntaktisch korrekte, Kombination möglich. Solche Kombinationen sind unter Umständen die einzige Möglichkeit komplexe Prozesspfade zu modellieren. Ein Beispiel für eine solche Kombination ist in Abbildung 5.29 dargestellt.

Das Beispiel ist Teil eines OP-Prozesses. Während der Anästhesist die Vitalparameter des Patienten dokumentiert, ergibt sich eine Komplikation bei der laufenden OP. Für ihn bedeutet das, dass er die Narkosedauer verlängern muss. Zudem muss er die Normalstation darüber in Kenntnis setzen, dass der Patient vorerst auf die Intensivstation verlegt wird. Selbige muss er ebenfalls über den ungeplanten Zugang informieren. In dieser Zeit kann er bereits beginnen den Intensivplan für den Patienten zu schreiben. In diesem Beispiel wurde die Möglichkeit, dass ein Patient

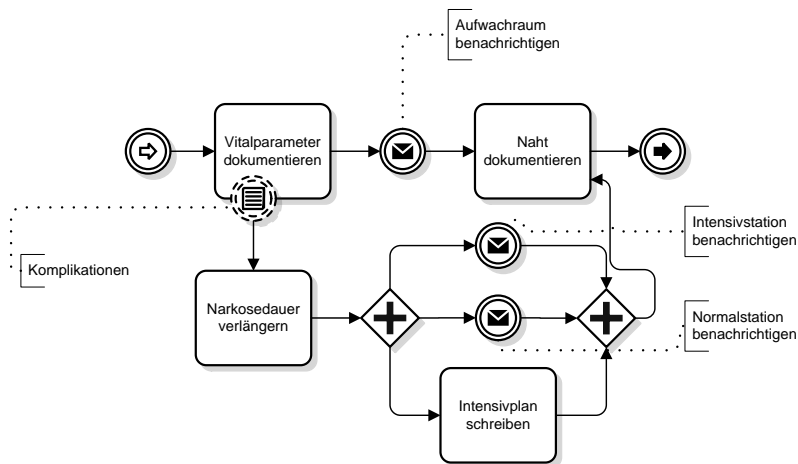


Abbildung 5.29: Kombination verschiedener Alternativen

auf Grund der Komplikationen die OP nicht überlebt ausgelassen. Dieser Fall müsste zusätzlich modelliert werden und würde damit die Komplexität des Modells weiter erhöhen. Diese Komplexitätserhöhung in Kombination mit den aufsummierten Schwächen der einzelnen Symbole machen, diese Alternative nicht allgemein anwendbar.

5.4 Vorschlag für neue Symbole

Die Aufteilung der vorgeschlagenen Symbole entspricht der im Standard verwendeten. Entsprechend dieser werden in diesem Abschnitt die Symbole eingeführt und deren Semantik ausführlich erläutert. Zudem soll nun an dieser Stelle noch einmal auf den Unterschied zwischen Ausnahmebehandlung und Kontextänderung eingegangen werden.

Den Anfang macht das Konzept der Ausnahmebehandlung, oft auch als *exception handling* bezeichnet. Dieses Konzept wird oft eingesetzt, um auf Fehler oder ungewollte Zustände in einem Programm zu reagieren. Diese Ausnahmen können entweder direkt an dem Ort bearbeitet werden, an welchem sie auftreten, oder an die nächst höhere Instanz weitergegeben werden. Beim Programmieren wäre dies zum Beispiel die aufrufende Methode.

Ein weiteres Prinzip der Ausnahmebehandlung ist das sogenannte *Sagas* (siehe [GMS87]). Es basiert auf dem „Alles-oder-Nichts“ Prinzip, welches auch bei Datenbanken Anwendung findet. Einzelne Aktivitäten werden als Teiltransaktionen betrachtet, bei welchen zwei Arten von Ausnahmebehandlung existieren. Welche davon angewandt wird, hängt davon ab, ob die Aktivität beendet oder abgebrochen wurde. Sollte eine Transaktion zum Zeitpunkt des Fehlers, bereits abgeschlossen worden sein, werden Kompensationsschritte für diese Transaktion eingeleitet. Andernfalls wird die Aktivität zurückgesetzt. Sagas findet vor allem im Bereich *Service-Oriented Computing* Anwendung.

Kontextänderungen hingegen sind nicht zwangsläufig Fehler oder unerwartete Situationen. Es

ist vielmehr das, was sich im Laufe der Prozessausführung ändert. An dieser Stelle muss jedoch darauf geachtet werden, welcher Kontext sich ändert. Als Kontext wird auch das Umfeld der Prozessausführung bezeichnet. Die Änderung dieses Umfelds, zum Beispiel der Ort an welchem der Prozess ausgeführt wird, kann allerdings auch eine Kontextänderung im Sinn dieser Arbeit herbeiführen. Wichtig hierbei ist auch, dass diese Änderung vom Kontext ausgeht.

Eine Kombination beider Konzepte ist durchaus möglich und praktikabel. Vor Allem die Anwendung von exception handling-Methoden im Rahmen der Kontextmodellierung.

Da nun die Abgrenzung geschaffen wurde, können die vorgeschlagenen Symbole vorgestellt und erläutert werden. Eine Übersicht dieser ist in Abbildung 5.30 dargestellt. Die Beschreibung erfolgt entlang der Abbildung von links nach rechts und von oben nach unten. Es wird jeweils ein Symbol vorgestellt, die Semantik erläutert und eine Abgrenzung zu bereits bestehenden Symbolen geschaffen. Zudem wird die Verwendung jedes Symbols mittels eines Beispiels verdeutlicht.

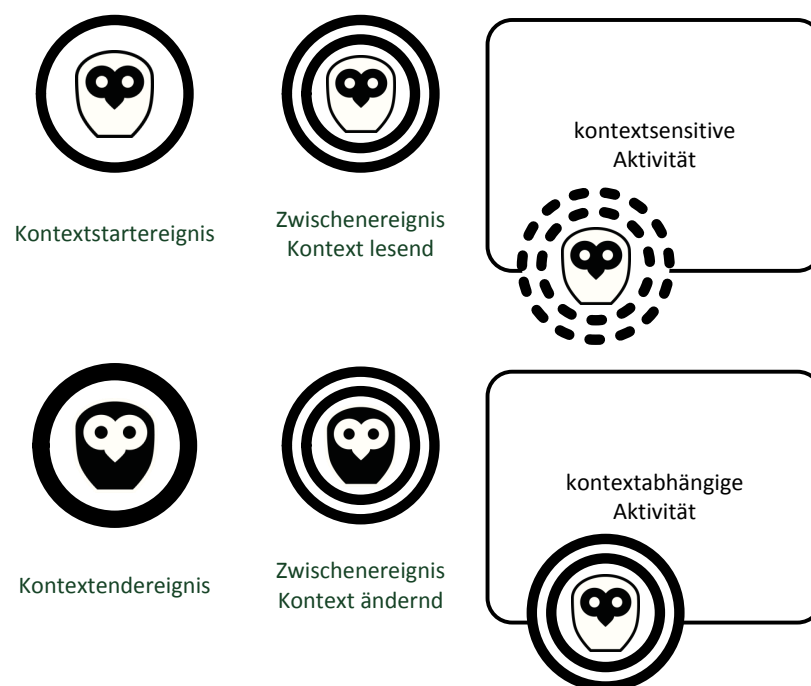


Abbildung 5.30: Vorgeschlagene Symbole

Den Anfang macht das Kontextstarterereignis, dieses ist eine Spezialisierung des normalen Starterereignisses. Es zeigt an, dass ein Prozess durch sich verändernden Kontext gestartet wird. Es wird empfohlen dieses Symbol mit einem zusätzlichen Kommentar zu versehen, um die Art der Kontextänderung zu spezifizieren. Ein Beispiel, wo spezialisierte Start- oder Endereignisse häufig angewendet werden, sind Subprozesse. Ein solches finden sie in Abbildung 5.31.

Dieser Subprozess könnte in einem größeren Software Engineering Prozess eingebettet sein und wird dadurch ausgelöst, dass ein Kollege sich krankmeldet. Die eingehende Krankmeldung bedeutet einen, zumindest zeitlich begrenzten, Personalausfall, was eine Kontextänderung be-

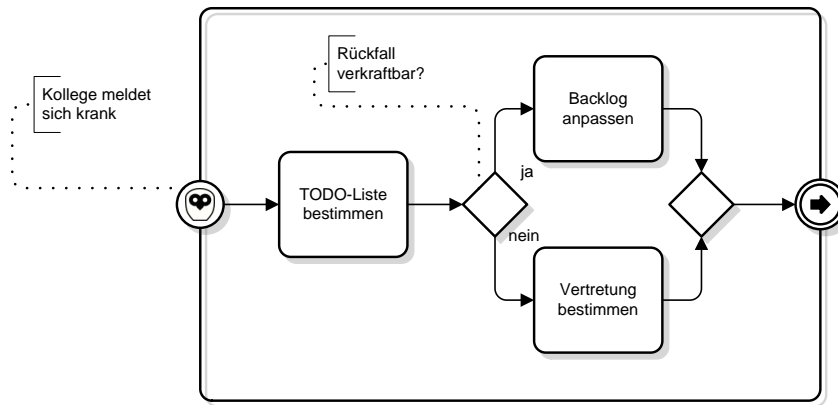


Abbildung 5.31: Beispiel für die Verwendung des Kontextstartereignisses

deuten kann. Nachdem die Krankmeldung eingegangen ist, wird die TODO-Liste des Kollegen für den Zeitraum, für den er sich krankgemeldet hat, überprüft, um zu entscheiden, ob der dadurch entstehende Rückschlag verkraftbar ist. Ist dies der Fall, muss der Backlog entsprechend angepasst werden. Falls nicht, so muss man einen qualifizierten Vertreter bestimmen.

Welchen Vorteil hat nun die Verwendung dieses Symbols entgegen dem bereits vorhandenen? Mit dem neuen Symbol soll eine klare Abgrenzung des Kontextes von einfachen Variablenänderungen sichtbar gemacht werden. In dem zuvor beschriebenen Beispiel wäre es zwar auch möglich gewesen ein Nachrichtenergebnis zu verwenden, jedoch wäre die Kontextänderung dadurch nicht mehr ersichtlich.

Das nächste wichtige Element ist das Kontextendereignis. Wie bereits das zuvor beschriebene, ist auch dieses eine Spezialisierung des normalen Endereignisses. Es wird verwendet, wenn der Abschluss eines Prozesses Auswirkung auf den Kontext hat. Dies kann sowohl ein planmäßiges als auch ein alternatives Ende sein. Dieses Ereignis ist besonders für Subprozesse von Interesse. Hier kann die Auswirkung des Prozesses auf den Kontext deutlich gemacht werden. Ein Beispiel für die Verwendung im Subprozess ist in Abbildung 5.32 zu finden.

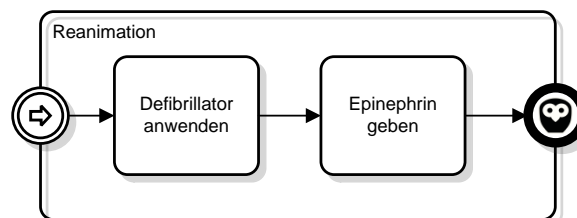


Abbildung 5.32: Beispiel für die Verwendung des Kontextendereignisses

Dieses Beispiel zeigt die letzten Aktivitäten des Reanimationsprozesses, wie er in einem Klinikum oder im Rettungsdienst vorkommen kann. Nach Anwendung des Defibrillators wird dem Patienten noch Epinephrin verabreicht, um ihn so zu stabilisieren. Je nachdem, wie der Patient reagiert, wird der auslösende Prozess beeinflusst. Diese Beeinflussung ändert den Kontext.

Ebenso wie das Kontextendereignis aufzeigt, dass das Prozessende den Kontext verändert, zeigt das Kontextstartereignis, dass der Prozess durch sich verändernden Kontext gestartet wird.

Ein Elementtyp, welcher in der BPMN ebenfalls große Bedeutung hat, sind die Zwischenereignisse. Hier wird zwischen ausgelöstem und eingetretenem unterschieden. In der vorliegenden Arbeit wird zunächst das eingetretene Zwischenereignis betrachtet.

Wie im Standard vorgesehen, wartet die Prozessausführung bei Ankunft an diesem Ereignis, bis es eintritt. Dieses Verhalten lässt sich gut an einem Beispiel erklären. Die nachfolgende Erklärung bezieht sich in Abbildung 5.33, welches ein Beispiel für die Verwendung dieses Ereignisses darstellt.

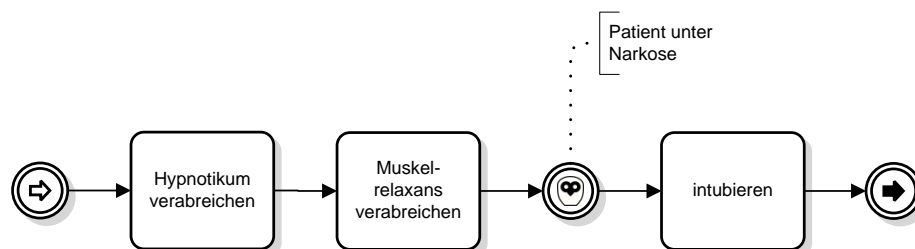


Abbildung 5.33: Beispiel für die Verwendung des Zwischenereignisses Kontext lesend

Der Patient erhält vom Anästhesisten ein Hypnotikum (Schlafmittel) und ein Muskelrelaxans verabreicht. Nun muss abgewartet werden, bis der Patient eingeschlafen ist, bevor man ihn intubieren kann.

Ähnliche Beispiele wie in Abbildung 5.33 lassen sich in jedem Fachbereich finden. Es ist jedoch auch möglich, ein Domänen unabhängiges Beispiel anzugeben: Die Verwendung des Kontextereignisses zur Steuerung eines ereignisbasierten Gateways. In diesem Szenario kann man zudem das Kontextereignis mit anderen Ereignistypen kombinieren, um so möglichst viele Fälle abzudecken.

Auch hier gilt wiederum, dass das Symbol genau die Nische auffüllt, welche die vorhandenen Symbole nicht abdecken.

Das ausgelöste Zwischenereignis hat eine etwas andere Semantik. Der Prozess löst das Ereignis aus, welches den Kontext ändert, und läuft weiter. Es muss nicht gewartet werden. In Abbildung 5.34 wird das Ereignis anhand eines Auszugs aus dem Prozess der Diagnoseerstellung exemplarisch verwendet. Zur Verdeutlichung der Semantik wird der Ausschnitt nun nochmals beschrieben.



Abbildung 5.34: Beispiel für die Verwendung des Zwischenereignisses Kontext schreibend

Sobald der Arzt die Laborergebnisse des Patienten ausgewertet hat, ändert das den Kontext. Die Änderung hierbei ist der Kenntnisstand, zuvor konnte der Arzt sich lediglich auf klar erkennbare Symptome und die Schilderung des Patienten stützen um eine Diagnose zu stellen. Die neuen Erkenntnisse stellen unter Umständen die Symptome und Schilderungen in einem neuen Licht dar und ermöglichen so eine Verfeinerung der Diagnose.

Ereignisse können in der BPMN allerdings nicht nur für sich allein stehen, sondern, wie bereits beschrieben, an Aktivitäten angehängt werden. Der Standard unterscheidet hierbei zwischen unterbrechenden und nicht unterbrechenden Ereignissen. Die entsprechenden Konzepte wurden ebenfalls auf die Betrachtung des Kontextes übertragen. Hierfür wurden im Grundlagenkapitel die Definitionen für Kontextsensibilität (Definition 2.4) und Kontextabhängigkeit (Definition ??) eingeführt. Entsprechend dieser Definitionen wurden auch neue Symbole entwickelt, welche an dieser Stelle erläutert werden.

Die kontextsensitive Aktivität muss, wie in der Definition beschrieben, bei einer Kontextänderung nicht unterbrochen werden. Eine mögliche Anwendung ist in Abbildung 5.35 gegeben, diese wird nun ebenfalls erläutert.

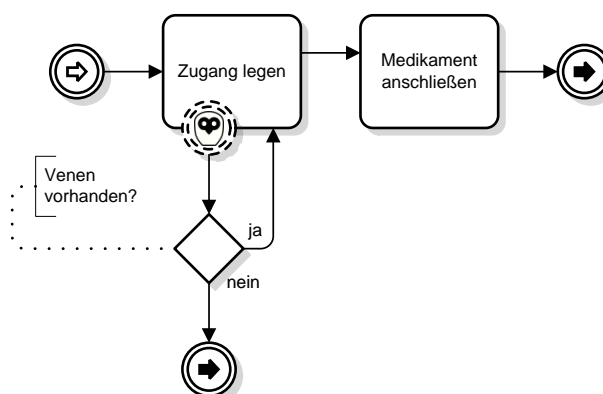


Abbildung 5.35: Beispiel für die Verwendung der kontextsensitiven Aktivität

Der Mediziner versucht einen venösen Zugang zu legen, was aus unbestimmten Gründen nicht klappt. Dieser Misserfolg hat eine Änderung des Kontext Patient zur Folge, da die Vene nun

nicht mehr verwendet werden kann. Solange der Patient noch über ausreichend zugänglicher Venen verfügt, kann folglich ein neuer Zugang gelegt werden, an welchem, im Anschluss daran, das Medikament angeschlossen wird. Der Arbeitsschritt muss bei einem Misserfolg jedoch nicht gleich unterbrochen werden, um, unter Umständen, etwas anderes zu tun. Ein solcher Prozessschritt wäre nach Definition kontextabhängig, exemplarisch in Abbildung 5.36 verwendet.

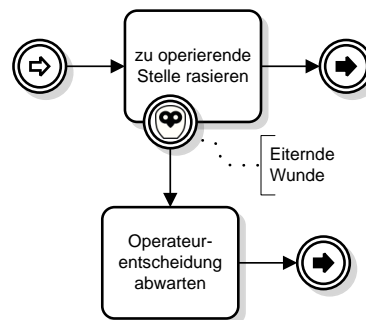


Abbildung 5.36: Beispiel für die Verwendung der kontextabhängigen Aktivität

Bei diesem Beispiel wurde dem Patienten die zu operierende Stelle rasiert. Hierbei kam eine eiternde Wunde zum Vorschein. In diesem Fall muss das Rasieren unterbrochen werden, da man durch die Rasur dem Patienten möglicherweise Schaden zufügen würde. Bevor das weitere Vorgehen besprochen werden kann, muss die Entscheidung des Operateurs abgewartet werden.

Nachdem die Beschreibung der Symbole erfolgt ist, bleibt die Frage zu klären, welchen Vorteil diese Symbole entgegen den bereits vorhandenen bringen. Um diese Frage beantworten zu können, muss eine klare Abgrenzung zu bereits vorhandenen gemacht werden. In Kapitel 5.3 wurden mögliche alternative Modellierungen diskutiert mit dem Ergebnis, dass keine davon optimal geeignet ist und jede über mehr oder weniger Schwachstellen verfügt. Die hier vorgestellten Symbole wurden auf Basis dieser Betrachtung verfeinert. Zuvor wurde eine Auswahl in Frage kommender Symbole erarbeitet. Diese stützten sich auf Beobachtungen reeller Prozesse und Diskussionen mit Fachanwendern. Das Ergebnis ist eine Menge von Symbolen, welche die Nische der Kontextmodellierung ergänzen soll. Sie ermöglichen die Modellierung von Kontextänderungen und den daraus resultierenden Aktivitäten. Hierfür ist es oftmals von Nutzen sie mit bereits vorhandenen Symbolen zu kombinieren, um alle Aspekte erfassen zu können. Die so erstellten Prozessmodelle sind kompakter und transparenter als bei der Modellierung mittels der diskutierten Alternativen.

5.5 Remodellierung mit neuen Symbolen

Im nun folgenden Abschnitt werden die zuvor vorgestellten Prozesse unter Verwendung der vorgeschlagenen Symbole neu modelliert. Die Modelle zielen vor allem darauf ab die kontextuellen Einflüsse auf die Prozesse und die Reaktionen darauf abzubilden. Die Aufteilung ist identisch zum bereits vorhandenen. Den Anfang machen die Prozesse aus der Domäne Medizin gefolgt von den Prozessen des Software Engineering.

5.5.1 Medizin

Bei den Prozessen dieser Anwendungsdomäne wurde, zusätzlich zu der Modellierung der kontextuellen Einflüsse, sofern möglich und sinnvoll Parallelität eingebaut. Zudem wurden die farbigen Markierungen, wie sie bei der Analyse verwendet wurden, aus den Modellen entfernt.

Narkoseeinleitung (Präoperative Phase)

Wie bereits zuvor macht der Prozess der Narkoseeinleitung den Anfang. Um die Modelle möglichst gut vergleichbar zu machen, wurde bei der Modellierung mit den vorgeschlagenen Symbolen die selbe Aufteilung gewählt wie bei der ersten Modellierung. Zusätzlich zur Kontextmodellierung wurden Schritte, bei denen es möglich und sinnvoll war, parallelisiert. Dieser Aspekt wurde in der ersten Modellierung herausgenommen, da es für die Analyse einfacher war die Schritte linear zu betrachten. Zudem hängt die Parallelisierung bei diesem Prozess davon ab wie viele Personen zum Anästhesie-Team gehören. Beide Prozessmodelle sind korrekt und lassen sich kombinieren. Die Aufteilung in die einzelnen Phasen ist identisch mit der ersten Modellierung.

- Phase 1

Gleich die erste Phase des Prozesses verfügt bereits über mehrere Anpassungen, zu sehen in Abbildung 5.37. Auffällig ist, dass die Prüfung der Patientenidentität hier eine eigener Subprozess ist. Dies ist nötig, um eine übersichtliche Kontextbehandlung modellieren zu können. Fasst man die Schritte nicht zusammen, müsste jedes eine eigene Kontextbehandlung erhalten. Dies widerspricht der Anforderung, dass die Modelle möglichst kompakt sein sollen. Durch das Zusammenführen in einen Subprozess ist dies jedoch möglich. Die Kontextbehandlung in diesem Fall ist die Suche nach der korrekten Patientenakte für den zu narkotisierenden Patienten. Sollte die Akte nicht auffindbar sein, kann der Patient aus Sicherheitsgründen nicht operiert werden und der Prozess wird beendet. Wird diese gefunden, muss nochmals die Identitätsprüfung durchgeführt werden. Der nächste Schritt, welcher neu modelliert wurde, ist die Prüfung der Nüchternheit des Patienten. Sollte dieser, entgegen aller Anweisungen, nicht nüchtern sein, führt dies zum Abbruch des Prozesses mit einem Vermerk in der Patientenakte. Da dieser nicht an eine bestimmte Person adressiert ist, kann er als Signal modelliert werden. Ebenfalls ist anzumerken, dass es keine feste Reihenfolge beim Anlegen der Vitalparameter Messgeräte gibt, weswegen diese

Schritte parallel modelliert wurden.

Bereits bei dieser verhältnismäßig kurzen Phase konnten durch die Verwendung der vor-

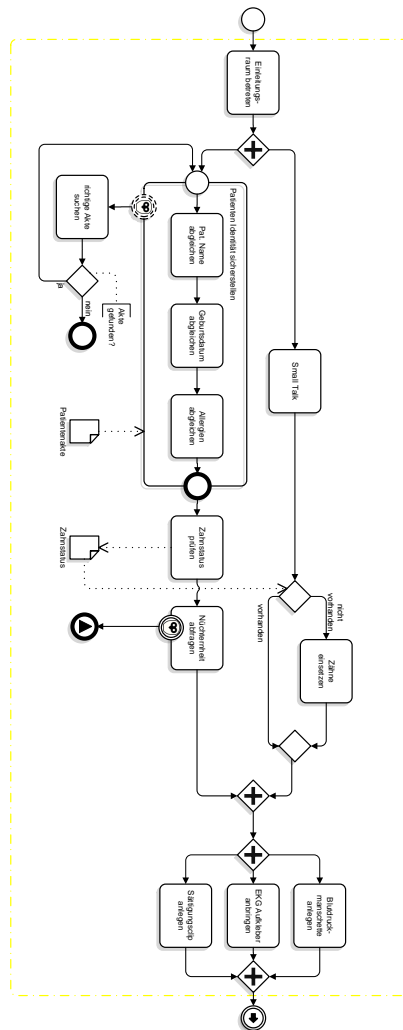


Abbildung 5.37: Narkoseeinleitung (Teil 1 von 6), Modellierung mit neuen Symbolen

geschlagenen Symbole, zusätzliche Informationen ins Prozessmodell eingefügt werden, ohne die Komplexität des Modells unnötig zu steigern. Durch die Kapselung der einzelnen Schritte zur Sicherstellung der Patientenidentität in einem Subprozess, ist es möglich den kontextuellen Einfluss aller drei Schritte auf einmal zu modellieren. Somit wird das Modell um die Information angereichert, welche Schritte zu unternehmen sind, sollte die Akte nicht zu dem aktuellen Patienten gehören, um ein Beispiel zu nennen.

- Phase 2

Die zweite Phase startet mit einer kontextsensitiven Aktivität, dem Legen eines venösen Zugangs. Wie bereits erläutert hängt der Erfolg dieser Aktivität vom Zustand der Patientenvenen ab. Ist es auf den ersten Versuch nicht möglich, wird solange eine neue Vene genommen, wie Venen vorhanden sind. Sollte keine Vene mehr zur Verfügung stehen, muss

ein arterieller Zugang gelegt werden. Unabhängig von der Art des Zugangs wird dem Patienten im nächsten Schritt eine Beatmungsmaske auf das Gesicht gehalten. Durch diese wird dem Patienten Narkosegas verabreicht, um die Einleitung zu beschleunigen und es dem Patienten somit einfacher zu machen. Da das zuvor genannte Gas jedoch nicht immer verabreicht wird, startet der Subprozess *Narkoseeinleitung* mit einem Kontextstartereignis. Ein weiterer Grund für die Verwendung dieses Ereignisses sind eventuelle Unverträglichkeiten beziehungsweise Allergien des Patienten. Diese können die Wahl der Medikamente, welche im Subprozess verabreicht werden, beeinflussen. Beendet wird der Subprozess mit einem Kontextendereignis. Abhängig von der Menge der verabreichten Medikamente ist der Patient für eine bestimmte Zeitdauer narkotisiert.

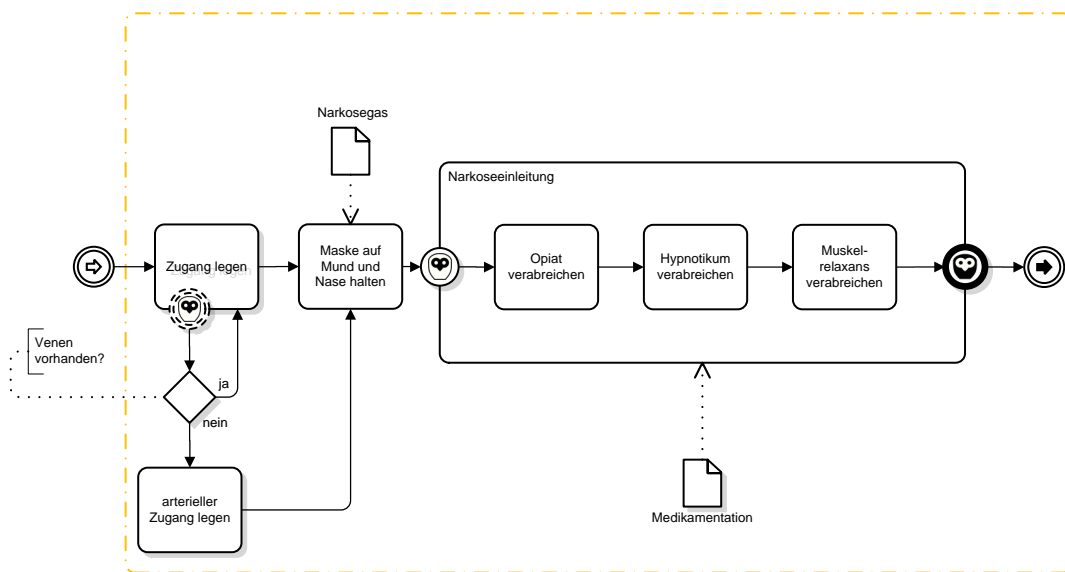


Abbildung 5.38: Narkoseeinleitung (Teil 2 von 6), Modellierung mit neuen Symbolen

Sobald der Patient schläft, kann der nächste Subprozess beginnen, welcher in Abbildung 5.39 sowie Abbildung 5.5 zu sehen. Im zweiten Abschnitt bestehen keine kontextuellen Abhängigkeiten, Daher wurde dieser nicht neu modelliert und wird in diesem Kapitel nicht nochmal abgebildet.

Die Änderungen beginnen hier etwa ab der Mitte des Subprozesses. Das Heben der Epiglottis des Patienten ändert den Kontext für den Arzt. Erst jetzt kann er beurteilen, ob er für das Legen des Tubus eine Einführhilfe verwenden möchte oder nicht. Abhängig von den Vorlieben des Arztes und den vorhandenen Möglichkeiten, kann nun, bei positiver Entscheidung, die Art der Einführhilfe gewählt werden. Nachdem diese Entscheidungen getroffen wurden, kann, als letzte Aktivität in diesem Abschnitt, die Magensonde gelegt werden. Da Erfolg oder Misserfolg dieser Aktion von der Speiseröhre des Patienten abhängt, ist diese Aktivität ebenfalls kontextsensitiv. Sollte man die Magensonde nicht gut einführen können, wird sie mittels der so genannten Magill-Zange eingeführt. Sobald die

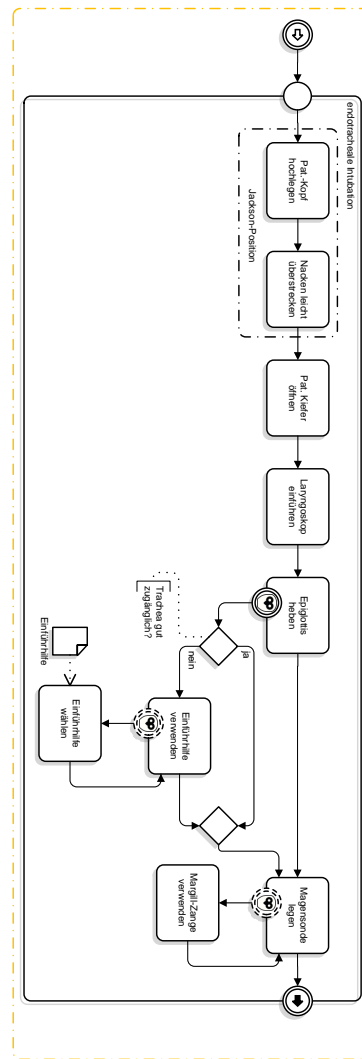


Abbildung 5.39: Narkoseeinleitung (Teil 3 von 6), Modellierung mit neuen Symbolen

Magensonde liegt, kann auch der Tubus eingeführt werden. Dieser Schritt liegt allerdings bereits im nächsten Abschnitt, welcher, wie zuvor erwähnt, nicht neu modelliert wurde.

Beim letzten Abschnitt dieser Phase, zu finden in Abbildung 5.40, existiert nur eine von Kontextänderungen betroffene Aktivität, die *engmaschige Kontrolle der Vitalparameter*. Sie ist eine klassisch kontextabhängige Aktivität, da eine Änderung des Kontext hier einen sofortigen Abbruch zur Folge hätte. Es besteht jedoch die Möglichkeit zu der Aktivität zurück zu kehren, je nach Ausgang des angeschlossenen Subprozesses. Die restlichen Aktivitäten dieser Phase sind kontextuell unabhängig und wurden deswegen in ihrer Modellierung nicht verändert.

- Phase 3

Die dritte Phase der Narkoseeinleitung ist, wie die Phase zuvor, auf mehrere Fragmente aufgeteilt. Das erste Fragment ist in Abbildung 5.40 zu finden (rote Markierung). Sie be-

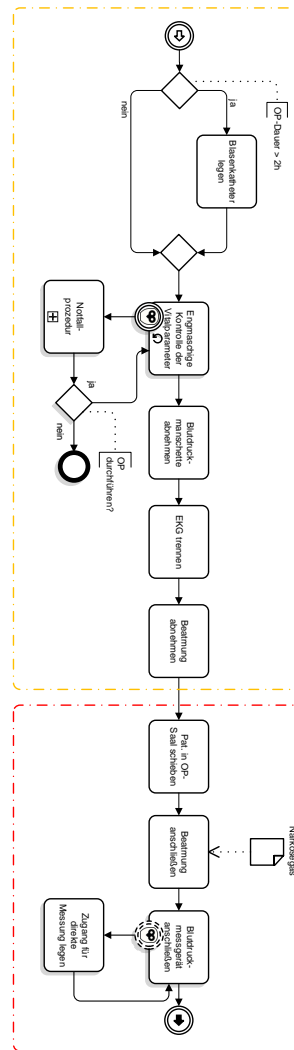


Abbildung 5.40: Narkoseeinleitung (Teil 5 von 6), Modellierung mit neuen Symbolen

gint wie gehabt damit, dass der Patient in den OP-Saal geschoben wird, wo er an die Beatmung angeschlossen wird. Die Neumodellierung erfolgte erst beim nächsten Schritt. Hier schießt man den Patient an die Blutdruckmessung an. Im Normalfall geschieht dies mittels einer Manschette. Sollte dies jedoch nicht möglich sein, muss eine direkte Blutdruckmessung über einen arteriellen Zugang gemacht werden. Sofern dieser noch nicht gelegt wurde, so muss dies an diesem Punkt nachgeholt werden.

Das nächste und zugleich letzte Fragment ist in Abbildung 5.41 zu sehen. Hier wurden drei kontextsensitive Aktivitäten neu modelliert. Die erste von ihnen ist das Anschließen der Medikation. Hier muss zum Einen beachtet werden, welche Medikamente bisher verabreicht wurden und zum Anderen, an welcher Stelle operiert wird. Wird auf Grund der zu operierenden Stelle ein starker Blutverlust für den Patienten erwartet, muss zusätzlich noch eine Kochsalzlösung angeschlossen werden. Die nächste neu modellierte Aktivität bezieht sich auf die Patientenlage während der OP. Bevor der Patient gelagert werden

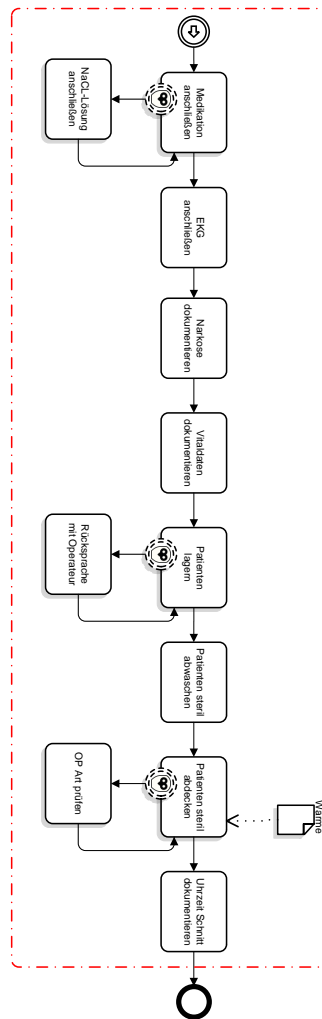


Abbildung 5.41: Narkoseeinleitung (Teil 6 von 6), Modellierung mit neuen Symbolen

kann, muss Rücksprache mit dem Operateur getroffen werden. Es muss nach Möglichkeit vermieden werden, dass die Arbeit des Operateurs durch die Lagerung des Patienten unnötig erschwert wird. Die letzte kontextsensitive Aktivität in diesem Prozess ist das sterile Abdecken des Patienten. Hier muss auf Grund der OP-Art entschieden werden, wie der Patient abgedeckt wird und wie viele Decken verwendet werden müssen damit der Patient nicht unnötig auskühlt.

Intra- und Postoperative Phase

Wie auch die präoperative Phase, wurden die intra- und Postoperative Phase mit den vorgeschlagenen Symbolen neu modelliert. Zusätzlich dazu wurden bei dieser Modellierung die Aktivitäten, bei denen es sinnvoll möglich war, parallel geschaltet. Dies geschah aus mehreren Gründen. Zum Einen war eine feste Reihenfolge der Aktivitäten für die Analyse einfacher und

zum Anderen, hat die Ausführungsreihenfolge keinen Einfluss auf den Kontext was die Modellierung dieser wiederum nicht nötig gemacht hat. An dieser Stelle dient die Parallelität dazu die Prozessmodelle kompakter zu gestalten.

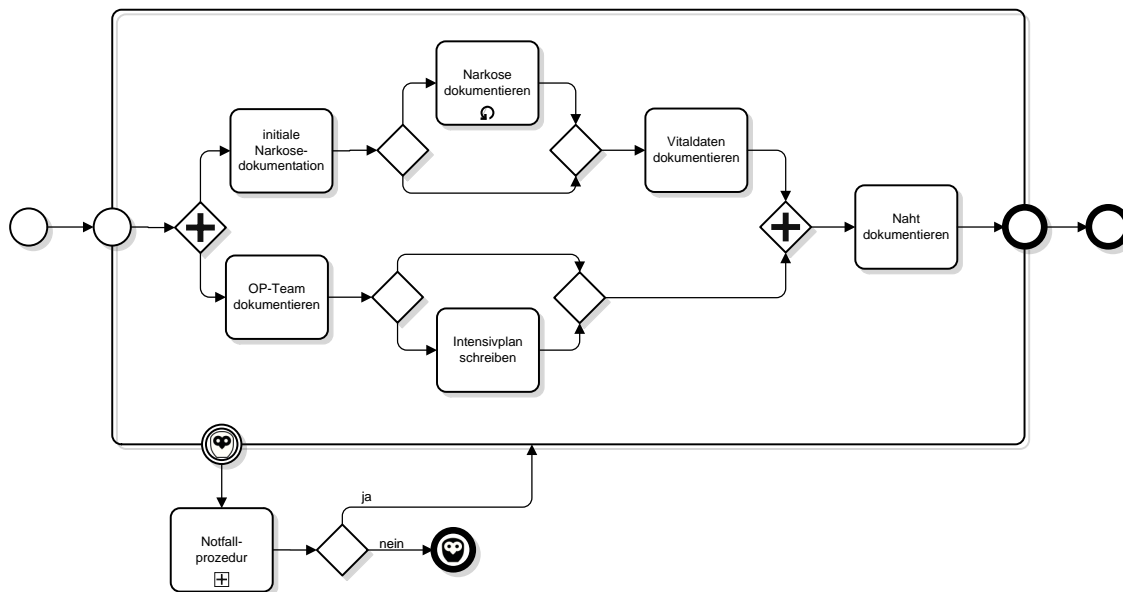


Abbildung 5.42: intraoperative Phase

Den Anfang macht, wie auch zuvor, die intraoperative Phase, zu sehen in Abbildung 5.42. Diese Phase wurde als Subprozess modelliert, um den wichtigsten kontextuellen Einfluss darstellen zu können: Den Notfall. Falls ein Patient während einer laufenden OP kollabiert oder ein anderer Notfall eintritt, so muss sofort auf diesen reagiert werden. Die Modellierung mittels Subprozess ermöglicht es den Kontexteinfluss sehr kompakt darzustellen, andernfalls müsste an jede Aktivität ein Ereignis angeheftet werden. Welche Notfallprozedur eingeleitet wird, hängt vom Notfall selbst ab. In manchen Fällen genügt es dem Patienten Epinephrin zu verabreichen, um ihn damit zu stabilisieren. Nach Abschluss der Notfallprozedur muss entschieden werden, ob die OP fortgesetzt werden kann. Diese Entscheidung hängt auch vom Ausgang der Notfallprozedur ab. Sofern der Patient zu diesem Zeitpunkt stabilisiert werden konnte, kann die OP fortgesetzt werden. Sollte dies nicht zutreffen oder der Patient sterben, wird die OP an dieser Stelle beendet. Der eigentliche Prozess wurde bei dieser Modellierung, wie bereits erwähnt, parallelisiert. Die Dokumentationsarbeit kann parallel erledigt werden, während die Vitalparameter überwacht werden.

Der nächste Phase der Operation ist die postoperative. Diese Phase findet, wie auch die präoperative, an zwei verschiedenen Orten statt. Dieser Ortswechsel ist kein Kontexteinfluss und wird daher in der Modellierung in Abbildung 5.43 lediglich als Gruppierung dargestellt. Diese dient rein der optischen Abgrenzung und der Übersichtlichkeit.

zweite Einfluss, welcher hier modelliert wurde, ist der Fall, dass der Patient kollabiert. In diesem Fall muss, wie auch während der intraoperativen Phase, die Bearbeitung der aktuellen Aktivität unterbrochen und der Patient stabilisiert werden.

5.5.2 Software Engineering

Die Prozesse des Software Engineering sind bereits sehr ausgereift, weswegen die neue Modellierung hier teilweise sehr knapp ausfällt.

Open Unified Process

Bei der Analyse des Open Unified Process wurde die Möglichkeit angesprochen, den Prozess auf zwei unterschiedliche Arten zu modellieren, wobei die Semantik der Modelle ebenfalls unterschiedlich ist. Im Folgenden werden beide Alternativen vorgestellt und auf deren Besonderheiten, sowie Stärken und Schwächen eingegangen. Die Reihenfolge der Varianten entspricht der Reihenfolge in der Analyse.

Die erste Möglichkeit, zu sehen in Abbildung 5.44, erlaubt kontextuellen Einfluss an zwei Stellen. Zu Beginn der Arbeit und am Ende. Die Modellierung beider Einflüsse wurde identisch realisiert, um sie möglichst generisch zu halten und die Anzahl der Möglichkeiten nicht einzuschränken. Zur Verdeutlichung der Semantik ist ein Beispiel eine gute Wahl.

Beispiel:

Im Rahmen dieses Beispiels soll ein Modul für ein Navigationssystem entwickelt werden. Das Modul soll den Kraftstoffverbrauch für die möglichen Routen berechnen und auf Basis der vorhandenen Routeninformationen die Optimale ermitteln. Als Implementierungssprache wurde C# gewählt. Variablen, welche aus mehreren Worten bestehen, sollen via Unterstrich verbunden werden. Weitere Einschränkungen wurden nicht getätigt. Das Entwicklerteam besteht aus drei Personen, welche bereits im Vorfeld am Endsystem gearbeitet haben. Nach Beginn des Lösungsentwurfs entschied die Projektleitung, dass die Bezeichnung der Variablen im CamelCase-Stil geschehen soll. Diese Kontextänderung ist an dieser Stelle trivial und kann geschehen, ohne dass die Arbeit unterbrochen werden muss. Anders verhält es sich jedoch, wenn anstatt der Variablenbezeichnung die Funktionalität des Moduls verändert wird. In diesem Fall wird der Prozess beendet und unter Umständen neu gestartet.

Ebenso verhält es sich bei der Integration der Lösung. Der Unterschied hier ist lediglich die Gewichtung der Kontextänderung. Während beim Lösungsentwurf die Änderung der verwendeten Programmiersprache noch verkräftbar ist, ist sie bei der Lösungsintegration nicht mehr möglich ohne die Lösung neu zu entwickeln, was jedoch einen Neustart des Prozesses zur Folge hätte.

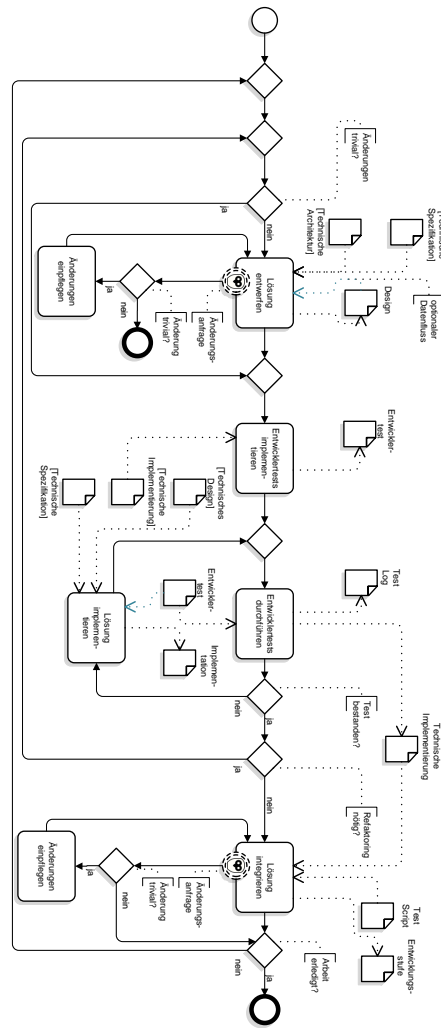


Abbildung 5.44: Develop Solution Increment Prozess (Kontextmodellierung Variante 1), Quelle: [Men14] mit eigenen Anpassungen

Wie man auch am Beispiel erkennen kann, wurden die Schritte im mittleren Bereich des Prozesses nicht neu modelliert. Der Grund hierfür ist, dass sie von Kontextänderungen nicht tangiert werden sollen. Sie verhalten sich ähnlich zum Sprint des Scrum-Prozesses und stellen, nach Ansicht des Autors, den kritischen Bereich dar. In diesem Bereich würde eine Kontextänderung den Abbruch des Prozesses zur Folge haben. Dieser Aspekt war es auch, der zu der zweiten Modellierung führe.

Wie bereits erwähnt, wurde bei der zweiten Modellierung, zu finden in Abbildung 5.45, versucht, den mittleren Bereich ebenfalls kontextuell abzudecken. Hierfür wurde der Prozess durch einen Subprozess gekapselt, welcher zudem die Kontextabhängigkeit modelliert. Sollte demnach bei einer der Aktivitäten eine Kontextänderung auftreten, wird diese durch den Subprozess abgefangen. Dieser wird dadurch jedoch unterbrochen. Dies ist nötig, um die Kontextänderung auszu-

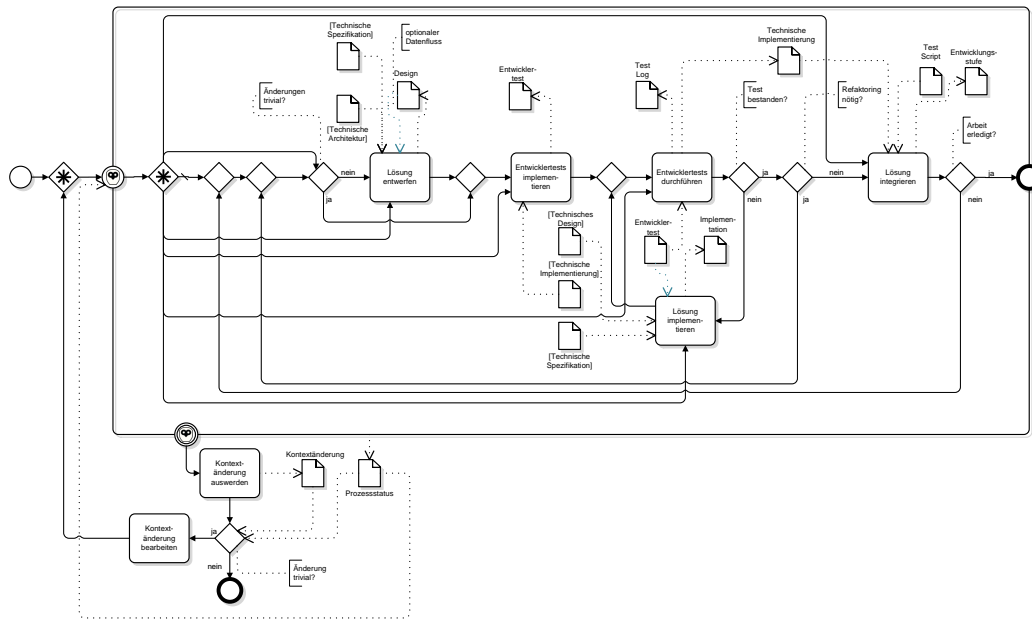


Abbildung 5.45: Develop Solution Increment Prozess (Kontextmodellierung Variante 2), Quelle: [Men14] mit eigenen Anpassungen

werten und unter Berücksichtigung dieser und des Status des Prozesses vor der Unterbrechung zu entscheiden, ob die Änderung trivial ist oder nicht. Hat die Änderung größere Auswirkungen, sprich sie ist nicht trivial, wird der Prozess an dieser Stelle beendet, um ihn anschließend eventuell neu zu starten. Falls die Änderung jedoch trivial sein sollte, kann sie entsprechend eingepflegt und der Prozess fortgeführt werden. Um den Einstieg an der korrekten Stelle zu gewährleisten, wurde das Startereignis des Subprozesses als Kontextstartereignis modelliert. Abhängig vom zuvor geschriebenen Prozessstatus steuert es das nachgeschaltete komplexe Gateway, um an die benötigte Stelle zu springen. Sollte der Prozess initial gestartet worden sein und daher der Prozessstatus noch den Initialwert haben, wird der default-Pfad gewählt. Dieser ist gekennzeichnet durch einen kleinen Querstrich durch den Pfad. Er entspricht dem normalen Prozessablauf wie im Kapitel 5.2 beschrieben. Jedoch weist diese Alternative eine eindeutige Schwäche auf. Durch das komplexe Gateway werden zusätzliche Prozesspfade eingefügt, was der Übersicht schadet. Zudem wird die aktuelle Aktivität immer unterbrochen, um die Kontextänderung zu analysieren. Auch wenn die Arbeit unter Umständen erneut aufgenommen wird, kann dieser Abbruch den Prozessfluss stören.

Scrum

Den Abschluss bei den neu modellierten Prozessen macht der Scrum-Prozess. Er hat eine kleine Besonderheit gegenüber den anderen Software Engineering Prozessen. Die Entwickler haben von vornherein einen Bereich angedacht, an welchem keine Beeinflussung stattfinden soll. Gemeint ist der Scrum Sprint. Aufgrund dieser Basisüberlegung wurde der Bereich nicht

neu modelliert. Um die Übersichtlichkeit des Modells zu verbessern, wird der Scrum Sprint als Subprozess behandelt. Die nachfolgende Beschreibung bezieht sich in Abbildung 5.46.

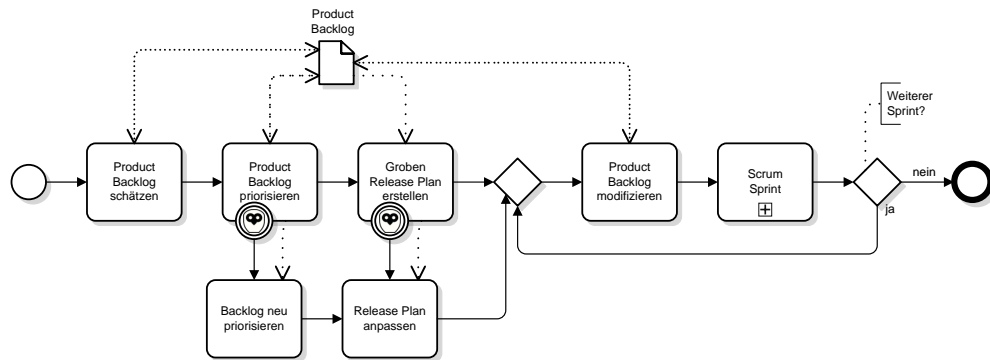


Abbildung 5.46: Scrum: Workflow mit Kontextmodellierung (verkürzt), Quelle: [Men14] mit eigenen Anpassungen

Entgegen der Analyse wurde der erste Prozessschritt nicht kontextabhängig oder kontextsensitiv modelliert. Der Grund hierfür ist, dass die Aktivität sehr unspezifisch ist und die einzige Veränderung, welche hier zu Buche schlagen würde, wäre der Abbruch des Projekts. Dies würde jedoch auch zum Abbruch des Prozesses führen. Beim darauf folgenden Schritt verhält es sich jedoch anders. Hier führt die Änderung des Kontext dazu, dass die Priorisierung überarbeitet werden muss. Falls dies geschieht, so muss immer auch der Release Plan entsprechend angepasst werden. Es ist jedoch auch möglich, dass die Kontextänderung erst bei der Erstellung des zuvor genannten Plans passiert. Hierbei muss der zuvor abgeschlossene Schritt nicht wiederholt werden. Weitere Neuerungen ergeben sich bei der Modellierung von Scrum nicht, da Scrum ein sehr flexibler Prozess ist. Besonders der Subprozess Scrum Sprint. Die Aktivitäten sind so gewählt beschrieben, dass sie auf Kontextänderungen reagieren können ohne zusätzliche Schritte einleiten zu müssen.

Die Prozessmodelle, die in diesem Kapitel, diskutiert wurden sind **vollständig** [A1]. Sowohl die klassische Modellierung in BPMN als auch die Kontextmodellierung der Prozesse. Die Prozesse enthalten ausreichend Informationen um die Prozesse zu verstehen ohne Fachwissen in den entsprechenden Anwendungsdomänen zu besitzen. Die Modelle aus den Kapiteln 5.1 und 5.2 wurden zudem ausschließlich mit den Symbolsatz der BPMN, ohne Erweiterungen, erstellt, was ebenfalls eine Anforderung war. Diese war eine der Anforderungen, welche von Fachanwendern aus den betrachteten Domänen ebenfalls für die Modelle geprüft und als erfüllt bestätigt wurde. Die Modelle des Kapitels 5.5 entsprechen streng genommen nicht den festgelegten Kriterien.

Alle in der vorliegenden Arbeit aufgezeigten und diskutierten Prozesse sind **syntaktisch korrekt** [A2], sie verstoßen an keinem Punkt gegen die BPMN 2.0 Syntax. Ebenso verhält es sich mit der Anforderung [A3], der **semantischen Korrektheit**. Jeder Prozess ist in sich geschlossen und spiegelt die Realität wieder. Dieses war das nächste Kriterium, welches von Fachanwendern nachgeprüft worden ist. Diese konnten die semantische Korrektheit der Modelle bestätigen. Die Prozesse werden deckungsgleich mit den hier aufgezeigten Modellen durchgeführt.

Die letzte der allgemeinen Anforderungen befasste sich mit der **Verständlichkeit** [A4] der Prozessmodelle. Hierbei wird gefordert, dass die Prozessmodelle derart gestaltet werden, dass Fachanwender diese erkenne und validieren können. Fachfremde sollen den Prozess nach kurzer Einarbeitungszeit verstehen und zusammenfassen können. Diese Anforderung wird ebenfalls von den Prozessmodellen erfüllt. Wenn auch der Prozess der *Narkoseeinleitung* ein verhältnismäßig lang ist und einen hohen Detailgrad besitzt, so ist er auf Grund der linearen Modellierung und gerade wegen der Vielzahl an Details für Fachfremde gut verständlich. Ebenso verhält es sich mit den Prozesse des Software Engineering. Hier ist der *OpenUP* Prozess für Fachfremde nicht sofort verständlich, benötigt jedoch keine lange Einarbeitungszeit.

Ebenso wie die allgemeinen Anforderungen, müssen auch die Anforderungen an die Erweiterung validiert werden.

Den Anfang macht dabei die syntaktische **Widerspruchsfreiheit** [E1]. Die in dieser Arbeit gewählte Erweiterung der BPMN waren neue Symbole. Diese Basieren dabei auf bereits Vorhan-

denem und fügen sich somit in die bestehende Palette der BPMN ein. Da die neu Modellierung der Betrachteten Prozesse aus den Kapiteln 5.1 und 5.2 in Kapitel 5.5 mit den bestehenden Symbolen in Kombination mit der Erweiterung geschah, sind diese Modelle syntaktisch widerspruchsfrei und verstoßen nicht gegen die BPMN Syntax.

Eine weitere Anforderung an die Erweiterung, war die **syntaktische Neuartigkeit** [E2], was im Endeffekt bedeutet, dass die Erweiterung nicht nur eine andere Verwendung von bereits bestehenden Symbolen sein muss. Die vorgeschlagenen Symbole basieren zwar auf der Symbolpalette der BPMN und besitzen daher syntaktische Überschneidungen mit dieser, sind aber dennoch nicht deckungsgleich mit den bestehenden Symbolen. Womit diese Anforderung erfüllt wurde.

Die Möglichkeit **kontextuelle Einflüsse und deren Auswirkungen auf Prozesse zu modellieren** [E3] war nicht nur eine Anforderung sondern auch eines der Ziele dieser Arbeit. Die Erfüllung dieser ist in Kapitel 5.5 zu finden. Hier wurden reelle Prozesse mittels der erweiterten Syntax modelliert und analysiert. Es wurden ausgewählte Kontextuelle Einflussfaktoren testweise in das Modell eingebaut. Das Ergebnis waren Prozessmodelle welche syntaktisch korrekt waren und dabei den Einfluss der Faktoren darstellten.

Als letzter Punkt ist die **Vergleichbarkeit** [E4] der Prozessmodelle zu diskutieren. Hierfür wurden den Fachanwendern die Prozesse aus den Kapitel 5.1, 5.2 sowie 5.3 vorgelegt, mit der Bitte diese gegeneinander abzugleichen. Bis auf die Tatsache, dass die Prozesse auf Grund der Kontextbetrachtung ein teilweise leicht verändertes Ausführungsverhalten hatten sie sonst identisch waren. Ein Beispiel für einen Prozess mit veränderten Ausführungsverhalten ist der OpenUP, bei der zweiten neu Modellierung wird ein zusätzliches Datenelement eingeführt, welches sich den Prozessstatus merkt, um im Fall einer kontextuellen Unterbrechung wieder bei der Aktivität fortfahren zu können, bei der abgebrochen wurde.

Verwandte Arbeiten

In diesem Kapitel soll die vorliegende Arbeit mit thematisch verwandten Arbeiten verglichen werden.

Ein Artikel, welcher sich mit der Verwendung von Kontext zur automatisierten Generierung von Prozessen beschäftigt ist *Contextual Generation of Declarative Workflows and their Application to Software Engineering Processes* [GOR11b]. Dieser Artikel stellt eine Methode vor um Prozessmodelle auf Basis von kontextuellen Merkmalen. Hierfür wird eine Menge möglicher Aktivitäten angegeben, aus welchen das System Untermengen aussucht und zur Ausführung bringt. Die Selektion basiert hierbei auf dem bereits erwähnten kontextuellen Einflüssen zum Zeitpunkt der Ausführung. Auf diese Art entsteht sukzessive ein, speziell auf die Lösung des vorhandenen Problems abgestimmtes, Prozessmodell. Der Unterschied zwischen diesem Artikel und der vorliegenden Arbeit ist, dass der Artikel Kontexteinflüsse nutzt um Prozessmodelle automatisiert zu generieren, während die Arbeit sich auf das Modellieren bereits erwähnter Einflüsse spezialisiert.

Ähnlich verhält es sich mit dem Artikel *Semantic Workflow Adaption in Support of Workflow Diversity* [GOR10], welcher sich mit automatisierter Adaption von Prozessmodellen an sich verändernden Kontext. Wie auch beim ersten vorgestellten Artikel liegt hier der Schwerpunkt auf der Verwendung von Kontext, zu Anpassung von Prozessmodellen, was thematisch nicht die Problemstellung dieser Arbeit abdeckt.

Ein Paper, welches eine etwas andere Richtung einschlägt ist *Event-driven Exception Handling for Software Engineering Processes* [GOR11a]. Das Paper beschäftigt sich mit der Verwendung von Kontext für komplexes Prozess-Exception-Handling. Die Exception-Handling Problematik wird zwar im Rahmen dieser Arbeit ebenfalls angeschnitten, ist jedoch nicht das eigentliche Thema.

Die nachfolgenden drei Paper befassen sich im Kern mit dem gleichen Thema: Der Verwendung von Kontext um automatisiert Datensammlungsprozesse zu generieren. Das erste der drei ist *Towards Process-based Composition of Activities for Collecting Data in Supply Chains*

[GMSR14b]. In diesem Artikel wird eine Annäherung an eine automatisierte und kontextuelle Zusammensetzung der Aktivitäten und Dienste durch einen spezifizierten Prozess.

Der Artikel *Towards Configurable Data Collection for Sustainable Supply Chain Communication* [GMSR14a] beinhaltet eine leichtgewichtige, automatisierte Annäherung an eine kontextuelle Prozesskonfiguration.

Im Paper *Challenges of Applying Adaptive Processes to Enable Variability in Sustainability Data Collection* [GMSR13] wird ein System beschrieben, welches Datensammelungsprozesse unterstützt.

Alle drei genannten Paper nutzen Kontexteinflüsse zur automatisierten Generierung und Unterstützung von Prozessen. Wie auch schon bei den Artikeln zuvor, wird bei diesen drei der Kontext verwendet um Prozessmodelle zu generieren. Während diese Arbeit sich ausschließlich um die Modellierung unter Verwendung der BPMN.

Eine Arbeit, welche sich ebenfalls mit der Erweiterung von BPMN beschäftigt ist *Analyse und Überführung von Softwareentwicklungsprozessen in die standardisierte BPMN Notation* [NTM14]. Die Arbeit befasst sich mit der Anwendungsdomäne Software Engineering und erweitert die BPMN um Möglichkeiten zur Darstellung von Zuständigkeiten und Abhängigkeiten sowie der Abbildung von Ergebnissen.

Die vorliegende Arbeit hat mit der zuvor genannten zwar deutliche Schnittpunkte (Software Engineering Prozesse, Erweiterung der BPMN), bewegt sich jedoch in eine andere Richtung. Bei [Men14] liegt der Schwerpunkt darauf Softwareentwicklungsprozesse in die BPMN Notation zu überführen, wofür selbige erweitert werden muss, während die vorliegende Arbeit sich mit der Modellierung von Kontext befasst.

Die Idee kontextuelle Änderungen zu betrachten und auf diese zu reagieren wird bereits im Gebiet Pervasive computing erforscht. Hierfür wurde einige kontextbezogene Frameworks vorgeschlagen. Diese Frameworks beschreiben die Implementierung von Diensten, welche ihr Verhalten an sich ändernden Kontext anpassen. Beispiel hierfür sind *Context Management* [JKM⁺03], *CASS* [FC04], *SOCAM* [GPZ04] und *CORTEX* [BC04].

Wie bereits erwähnt, dienen diese Frameworks in erster Linie dazu, Anwendungen und Dienste an sich verändernden Kontext anzupassen, während diese Arbeit sich mit Kontexteinflüssen auf Prozesse und der Möglichkeit diese zu Modellieren befasst.

Es gibt nur sehr wenig Arbeiten, welche die Kombination aus Prozessmanagement und Kontext zum Thema haben. Eine davon ist *inContext* [DD07]. Allerdings liegt der Fokus dieser Arbeit auf dem Teamwork und wiederum nicht auf der Modellierung kontextueller Einflüsse, was in der vorliegenden Arbeit das Hauptthema darstellt.

Zusammenfassung, Fazit und Ausblick

Hier wird das zuvor erarbeitete zusammengefasst, wobei Besonderheiten der Arbeit nochmals hervorgehoben werden. Im Anschluss an die Zusammenfassung wird ein kurzer Ausblick auf mögliche zukünftige Projekte und Entwicklungen gegeben. Um die Arbeit abzurunden, wird in diesem Kapitel noch das Fazit gezogen.

8.1 Zusammenfassung und Fazit

Zusammenfassend lässt sich feststellen, dass die Modellierung kontextueller Einflüsse nicht nur ein theoretisches Thema ist, sondern auch eine hohe Praxisrelevanz besitzt. Um zu diesem Punkt zu kommen, muss man jedoch zuvor die einzelnen Kapitel der vorliegenden Arbeit nochmals betrachten.

Um Kontexteinflüsse auf Prozesse modellieren zu können, muss man sich klar machen was genau Kontext im Allgemeinen und im Prozessmanagement im Speziellen bedeutet. Außerdem muss eine Abgrenzung zwischen der Reaktion auf eine Kontextänderung und einer Kompensation geschaffen werden. Ist dies geschehen, ist es wichtig Anforderungen an die Modellierung zu formulieren. Diese sollen zum Einen sicherstellen, dass die Modelle vergleichbar sind und zum Anderen die syntaktische Korrektheit der Modelle stets gewährleistet. Mit diesem Rüstzeug lassen sich nun via Fallstudien Beispielprozesse aus zwei Anwendungsdomänen erstellen. Die Analyse dieser Prozesse liefert wichtige Einblicke in die Möglichkeiten kontextueller Einflüsse. Gleichzeitig sieht man an den Analysen, dass es schwer ist, Kontext adäquat zu modellieren. Daher ist es von hoher Bedeutung alle möglichen Varianten, welche die BPMN beinhaltet, zu untersuchen. Jede dieser Alternativen wurde in der vorliegenden Arbeit auf ihre Eignung hin untersucht. Sofern es sinnvoll war, wurde die Schwäche jeder Alternative mit einem Beispiel aufgezeigt.

Aus dem zuvor genannten wurde ein Vorschlag für neue Symbole ausgearbeitet. Die einzelnen Symbole des Vorschlags wurden beschrieben und die Verwendung dieser mittels Beispielen näher erläutert. Um die Symbole einer Art Praxistest zu unterziehen, wurden die Prozesse, welche

zuvor aufgenommen und analysiert wurden, neu modelliert. Bei dieser Modellierung wurden die vorgeschlagenen Symbole in Kombination mit einigen der zuvor diskutierten Alternativen verwendet. Auch diese Modelle wurden wiederum analysiert unter Hervorhebung ihrer Besonderheiten. Zum Abschluss erfolgte eine Validierung der Arbeit, auf Basis der in Tabelle 3.1 notierten Anforderungen. Hierfür wurde jede der beschriebenen Anforderungen nochmals betrachtet und retrospektiv mit der Arbeit verglichen.

Kurz gefasst kann man sagen, dass es möglich ist kontextuelle Einflüsse auf Prozesse zu modellieren. Jedoch sollte darauf geachtet werden nicht zu viel und zugleich nicht zu wenig zu modellieren. Diese Kurzfassung soll nun aufgeschlüsselt werden.

Es ist möglich kontextuelle Einflüsse auf Prozesse zu modellieren. Für sich allein hat der Satz mehrere Interpretationsmöglichkeiten. Wie in der Arbeit aufgezeigt wurde, ist es durchaus möglich Kontext nur unter Verwendung von BPMN eigenen Symbolen zu modellieren. Dies ist allerdings nur eingeschränkt machbar, da jede Möglichkeit über gewisse Schwächen verfügt. Für kleine Modelle oder welche mit wenig Kontexteinfluss ist dies jedoch der Einführung zusätzlicher Symbole vorzuziehen. Sobald das Modell allerdings einen bestimmten Schwellwert überschreitet, ist es empfehlenswert die Symbolpalette der BPMN zu erweitern. Dieser Schwellwert ist allerdings nicht fest, sondern abhängig vom Fachgebiet und der Komplexität des Prozesses.

Man sollte nicht zu viel und nicht zu wenig modellieren. Zugegebenermaßen ist es nicht einfach herauszufinden, wie viel zu viel ist. Auf jeden Fall ist es nicht ratsam wirklich jeden noch so kleinen Einfluss zu modellieren. Das daraus resultierende Ergebnis ist weder besonders übersichtlich noch zielführend. Einflussfaktoren können unter Umständen zu allgemeineren Faktoren zusammengefasst werden. Dies ist vor allem in den Fällen möglich, in welchen die Reaktion auf die Änderung gleich oder zumindest sehr ähnlich ist. Ebenso muss man sich überlegen, ob wirklich jeder Einfluss Auswirkungen auf den Prozess hat und ob diese nicht innerhalb der einzelnen Aktivitäten abgefangen werden können. Diese Überlegungen helfen die Anzahl der zu modellierenden Einflüsse zu regulieren und damit das Prozessmodell schlanker und übersichtlicher zu gestalten.

Die vorliegende Arbeit versucht einen Schritt in die beschriebene Richtung zu machen, indem der Symbolsatz der BPMN erweitert wird.

8.2 Ausblick

Die Möglichkeit der Modellierung von Kontexteinflüssen auf Prozesse ist, nach Ansicht des Verfassers der vorliegenden Arbeit sowie einiger Fachanwender, ein Themengebiet, welches an Bedeutung weiter zunehmen wird. Die Kenntnis dieser Einflüsse ist sowohl für Dokumentations- als auch für Schulungszwecke hilfreich. Auch ist es denkbar diese Einflüsse in die automatisierten Prozessausführung einfließen zu lassen. Das Wissen um diese Einflüsse und deren Auswirkungen, ermöglicht eine frühzeitige und detaillierte Planung. Daraus resultieren schnelle-

re Reaktionszeiten, was zu finanziellen Einsparungen führt.

Dies sind nur einige wenige Beispiele für mögliche zukünftigen Entwicklungen.

Literaturverzeichnis

- [ADB⁺99] ABOWD, Gregory D. ; DEY, Anind K. ; BROWN, Peter J. ; DAVIES, Nigel ; SMITH, Mark ; STEGGLES, Pete: Towards a better understanding of context and context-awareness. In: *Handheld and ubiquitous computing* Springer, 1999, S. 304–307
- [All09] ALLWEYER, Thomas: *BPMN 2.0-Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung*. BoD–Books on Demand, 2009
- [BBB⁺01] BECK, Kent ; BEEDLE, Mike ; BENNEKUM, Arie van ; COCKBURN, Alistair ; CUNNINGHAM, Ward ; FOWLER, Martin ; GRENNING, James ; HIGHSMITH, Jim ; HUNT, Andrew ; JEFFRIES, Ron ; KERN, Jon ; MARICK, Brian ; MARTIN, Robert C. ; MELLOR, Steve ; SCHWABER, Ken ; SUTHERLAND, Jeff ; THOMAS, Dave: *Manifesto for Agile Software Development*. <http://www.agilemanifesto.org/>. Version: 2001
- [BC04] BIEGEL, Gregory ; CAHILL, Vinny: A framework for developing mobile, context-aware applications. In: *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on* IEEE, 2004, S. 361–365
- [DD07] DORN, Christoph ; DUSTDAR, Schahram: Sharing hierarchical context for mobile web services. In: *Distributed and Parallel Databases* 21 (2007), Nr. 1, S. 85–111
- [Ecl] ECLIPSE FOUNDATION: *OpenUP*
<http://epf.eclipse.org/wikis/openup/>.
- [FC04] FAHY, Patrick ; CLARKE, Siobhan: CASS—a middleware for mobile context-aware applications. In: *Workshop on Context Awareness, MobiSys* Citeseer, 2004
- [FR12] FREUND, Jakob ; RÜCKER, Bernd: *Praxishandbuch BPMN 2.0*. Carl Hanser Verlag GmbH Co KG, 2012
- [GMS87] GARCIA-MOLINA, Hector ; SALEM, Kenneth: Sagas. In: DAYAL, Umeshwar (Hrsg.) ; TRAIGER, Irving L. (Hrsg.): *SIGMOD Conference*, ACM Press, 1987, 249-259

- [GMSR13] GRAMBOW, Gregor ; MUNDBROD, Nicolas ; STELLER, Vivian ; REICHERT, Manfred: Challenges of Applying Adaptive Processes to Enable Variability in Sustainability Data Collection. In: *3rd Int'l Symposium on Data-Driven Process Discovery and Analysis (SIMPDA'13)*, CEUR-WS.org, August 2013 (CEUR Workshop Proceedings 1027), 74–88
- [GMSR14a] GRAMBOW, Gregor ; MUNDBROD, Nicolas ; STELLER, Vivian ; REICHERT, Manfred: Towards Configurable Data Collection for Sustainable Supply Chain Communication. In: *CAiSE'14 Forum*, CEUR-WS.org, June 2014 (CEUR Workshop Proceedings)
- [GMSR14b] GRAMBOW, Gregor ; MUNDBROD, Nicolas ; STELLER, Vivian ; REICHERT, Manfred: Towards Process-based Composition of Activities for Collecting Data in Supply Chains. In: *6th Central European Workshop on Services and their Composition (ZEUS 2014)*, 2014
- [GOR10] GRAMBOW, Gregor ; OBERHAUSER, Roy ; REICHERT, Manfred: Semantic Workflow Adaption in Support of Workflow Diversity. In: *4th Int'l Conf. on Advances in Semantic Processing (SEMAPRO'10)*, Xpert Publishing Services, October 2010, 158–165
- [GOR11a] GRAMBOW, Gregor ; OBERHAUSER, Roy ; REICHERT, Manfred: Contextual Generation of Declarative Workflows and their Application to Software Engineering Processes. In: *Int J on Advances in Intelligent Systems 4* (2011), Nr. 3&4, 158–179.
<http://dbis.eprints.uni-ulm.de/795/>
- [GOR11b] GRAMBOW, Gregor ; OBERHAUSER, Roy ; REICHERT, Manfred: Event-driven Exception Handling for Software Engineering Processes. In: *5th Int'l Workshop on Event-driven Business Process Management (edBPM'11), BPM'11 Workshops*, Springer, August 2011 (LNBIP 99), 414–426
- [GPZ04] GU, Tao ; PUNG, Hung K. ; ZHANG, Da Q.: A middleware for building context-aware mobile services. In: *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th Bd. 5 IEEE*, 2004, S. 2656–2660
- [JKM⁺03] JANI, M ; KELA, Juha ; MALM, Esko-Juhani u. a.: Managing context information in mobile devices. In: *IEEE pervasive computing 2* (2003), Nr. 3, S. 42–51
- [LAR] PFLEGEWIKI
<http://www.pflegewiki.de/wiki/Laryngoskop>
- [Men14] MENHORN, Nicole ; GRAMBOW, Gregor (Hrsg.) ; REICHERT, Manfred (Hrsg.): *Analyse und Überführung von Softwareentwicklungsprozessen in die standardisierte BPMN Notation*. <http://dbis.eprints.uni-ulm.de/1048/>. Version: April 2014
- [NTM14] NEUS, Sebastian ; TROMPETER, Jens ; MANDISCHER, Martin: *Einführung in Scrum - Scrum Kompakt*. <http://scrum-kompakt.itemis-hosting.de/>. Version: 2014

- [OMG] OBJECT MANAGEMENT GROUP (OMG) <http://www.omg.org>
- [OMG11] OMG: *Business Process Model and Notation (BPMN), Version 2.0*. <http://www.omg.org/spec/BPMN/2.0>. Version: January 2011
- [Rei13] REICHERT, Manfred: *Business Process Management*. Folientexte zur Vorlesung, WS13/14, Universität Ulm (Fakultät für Informatik), 2013
- [Sil09] SILVER, Bruce: *BPMN method and style*. Bd. 2. Cody-Cassidy Press Aptos, 2009
- [Too] PROZESSMANAGEMENT BLOG:
Übersicht über BPMN Tools zur Prozessmodellierung
<http://prozessmanagement-blog.ch/post/32587959467/uebersicht-bpmn-prozessmodellierungs-tools>
- [WHO] WORLD HEALTH ORGANISATION:
Surgical Savity Checklist (first edition) http://www.who.int/patientsafety/safesurgery/tools_resources/SSSL_Checklist_finalJun08.pdf?ua=1

Name: Andreas Eichwald

Matrikelnummer: 668749

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Andreas Eichwald